100% AMIGA

Amazing COMPUTING™

Your Original AMIGA® Monthly Resource

For The Commodore AMIGA®

Volume 5 No. 1 January 1990
US $3.95 Canada $4.95

AMIGA ANIMATION

AMIGA ANIMATION
AMIGA ANIMATION
AMIGA ANIMATION
AMIGA ANIMATION
AMIGA ANIMATION
AMIGA ANIMATION
AMIGA ANIMATION
AMIGA ANIMATION
AMIGA ANIMATION
AMIGA ANIMATION
AMIGA ANIMATION
AMIGA ANIMATION
AMIGA ANIMATION
AMIGA ANIMATION
AMIGA ANIMATION
AMIGA ANIMATION
AMIGA ANIMATION
AMIGA ANIMATION
AMIGA ANIMATION

Sculpt-Animate 4D

Animation in BASIC

Anim-Aids

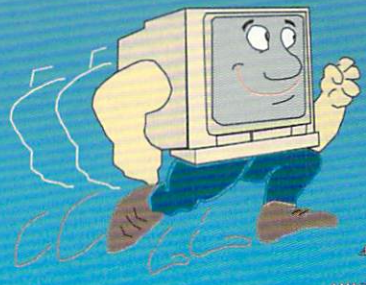
The 1989 BADGE Killer Contest Winners

AMIGA Programming:

Scanning The Screen, Fractals Part Four

Calculating the Wind Chill Temperature

Menu Builder



Plus:
Thinker

Hardware Home Control

01

0 74470 74710 9

Amazing COMPUTING™ For The Commodore AMIGA

Your Original AMIGA® Monthly Resource

Volume 5, Number 1
January, 1990

ANIMATION ON THE AMIGA

Dear Animators 41
by Barry Solomon
Introducing AC's Animation section.

Ladies And Gentlemen, we give you... THE WINNER! 42
by Bradley W. Schenck
The making of the 1989 BADGE Killer Demo Contest winner, *The Sentinel*.

Animation For Everyone 47
by Barry Solomon
Animation Can Be a Moving Experience.

PIC-MAGIC: A Product Profile 54
by Orlando Furioso
Wonderfully sketched, ready-to-color images!

Sidebar
Brushing-Up On Animation 54
by Barry Solomon
Cut, paste, and animate your way to fame.

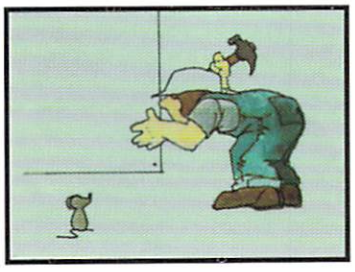
Animation With Sculpt-Animate 4D 51
by Lonnie Watson
Create some stunning artwork.

Animation? BASICally! 58
by Mike Morrison
Using Cell animation in AmigaBASIC.

Anim-Aids 64
by Barry Solomon
Where to find the help you need.

PROGRAMMING

Menu Builder 23
by Tony Preston
Building menus with Intuition.



Facing the CLI 39
by Mike Morrison
Disk structures and startup-sequences.

Dual Demo 66
by Thomas Eshelman
Programming an arcade game can be easy.

Scanning The Screen 82
by Paul Castonguay
Part Four in the fractals series.

It's Colder Than You Think 87
by Robert Klimaszewski
A simple program to calculate the wind chill temperature.

All program listings are available on AC Disks!

REVIEW

Thinker 90
by Robert Klimaszewski
"An Idea Processor".

DEPARTMENTS

Feedback Forum 4

Users Groups 21

Index of Advertisers 96

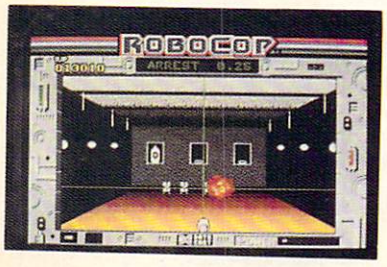
Public Domain Software 109

•CONTENTS•

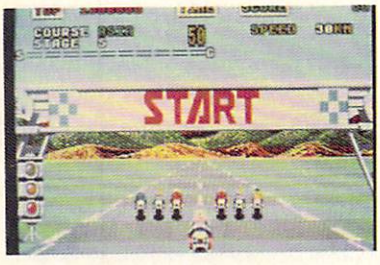
COLUMNS

New Products and Other Neat Stuff 8
by Elizabeth G. Fedorzyn
Pro Tennis Tour, Keef The Thief, and Distant Suns top the list.

Snapshot 15
by R. Bradley Andrews
Three arcade favorites, and a space role-playing game.



RoboCop *- An arcade game based in the future.*



Super Hang-On *- Only your skill and reflexes will allow you to win the ultimate race.*

Video Schmideo 18
by Barry Solomon
There's no business like the Video Business.

No Fishing! 27
by Graham Kinsey
A preview of Workbench 1.4. Plus a fantastic gambling game.

Bug Bytes 33
by John Steiner
More bugs and upgrades for the Amiga user.

Roomers 35
by The Bandito
The Bandito's fearless predictions for 1990 are here.

PD Serendipity 56
by Mike Morrison
Animations from the '89 BADGE Killer Demo contest.

LaserBoing *- Mike van der Sommen's entry to the 1989 BADGE Killer Demo Contest. "This 'Amiga Boing' ball is just loaded with energy."*

Amiga Circuits 98
by John Iovine
Controlling large amounts of electrical energy.

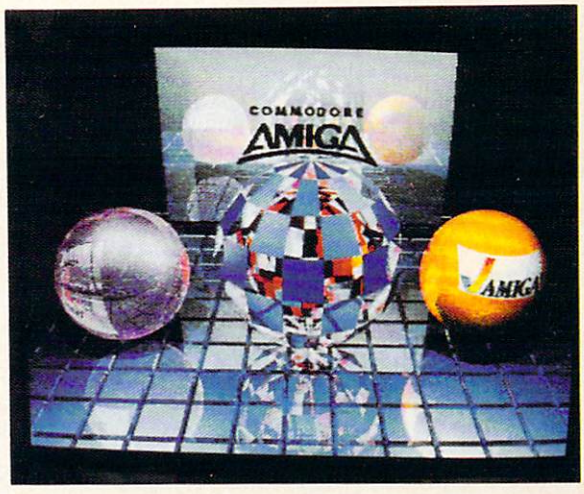


C Notes From The C Group 104
by Stephen Kemp
Recursive functions in C.

Guest Commentary

More Advice For A1000 Upgrades 6
Richard Taylor, President of Expansion Technologies speaks out.







Cover
Illustration by Brian Fox
Graphic Design by Ernest P. Viveiros Sr.

VIRUS ALERT!

AC's Technical Editor Explains How To Handle A New Virus.

by Michael Morrison

This is the type of article I hate to write. There is yet another virus going around. The virus is called Xeno and is not a bootblock virus but a file virus. This means that it attaches itself to executable files such as commands or even programs. When you use the command, the virus puts itself in memory and attaches itself to any other executable file that is used.

There are several ways to recognize whether or not your computer is infected with the Xeno virus:

–The commands in the C directory are 1124 bytes longer.

–The date on an executable file has been changed to a recent date.

–Your computer crashes when you try to print to the parallel port.

– You get the 'file not an object module' error when you try to use common C commands (dir, cd, assign, etc.).

Unfortunately, there has been no way to detect this virus until recently. We picked it up, here at AC, about a month ago, but none of our virus checkers could detect it. If you have ordered AC #5 or AC #6, they are infected with the virus. Please use one of the methods described here to disable the infected files, or return the disks to us, and we will send you clean disks.

There are a few programs available that will help rid your system of the Xeno virus. Steve Tibbet has just finished VirusX 4.0 which is available on BBS's or on our inNOCKulation disk (see ad page 86). VirusX 4.0 will detect the Xeno virus in memory, but it can not help any files that may be infected. Steve has also included a program called KV. This program will disable any files infected with the Xeno virus, but it does not remove the virus from the files. The virus intertwines itself with a file, making it difficult to remove. Steve is working on a version of KV that will actually remove the virus from the infected files.

Another program also available on BBS's or on our inNOCKulation disk is called XenoZap, by Kevin Kelm. Kevin wrote XenoZap after both of our computers were infected, and I asked him for help. XenoZap is similar to KV, except that it will search an entire device for the virus, in all directories and disable it in all infected files. This means you can type:

```
df0:XenoZap dh0: df0: df1: df2: XenoZap
```

from the CLI (if XenoZap is on the disk in df0:). XenoZap will first check memory for the virus and disable it (if present), and then check every file, in every directory, on dh0:, df0:, df1:, df2: and disable the virus in all infected files.

Once XenoZap is used to check the system and its files, it has become infected with the virus. This means that if the program does not check for and disable the virus on itself, the next time the program is run, it will infect every file that it is currently checking. Therefore, it is important to include XenoZap at the end of the list of items that XenoZap will be checking. This should eliminate the virus from XenoZap and eliminate the virus from your system. Remember to check all of your floppies!

These people who have nothing better to do but create damaging viruses really baffle me. It is unfortunate and sad. I am only thankful that talented people like Steve Tibbet, Kevin Kelm, and others are willing and able to spend their time and resources to combat these viruses. My hat is off to all of them. Both VirusX and XenoZap are public domain. This means they are free to everyone. Even though the authors don't request any gratuities for their work, please send them something if you use either of these programs. This will encourage them to continue their efforts in this battle. I hope this article will help save some of you from some headaches.

•AC•

Amazing COMPUTING™ — Your Original AMIGA® Monthly Resource — For The Commodore/Amiga

Amazing Computing For The Commodore AMIGA™

ADMINISTRATION

Publisher:	Joyce Hicks
Assistant Publisher:	Robert J. Hicks
Circulation Manager:	Doris Gamble
Asst. Circulation:	Traci Desmarais
Asst. Circulation:	Donna Viveiros
Corporate Trainer:	Virginia Terry Hicks
Traffic Manager:	Robert Gamble
International Coordinator:	Marie A. Raymond
Marketing Manager:	Ernest P. Viveiros Sr.

EDITORIAL

Managing Editor:	Don Hicks
Associate Editor:	Elizabeth Fedorzyn
Hardware Editor:	Ernest P. Viveiros Sr.
Technical Editor:	J. Michael Morrison
Music & Sound Editor:	Richard Rae
Video Editor:	Barry Solomon
Copy Editor:	Aimée B. Abren
Copy Editor:	Derek J. Perry
Copy Editor:	Troy J. Thomas
Art Director:	William Fries
Photographer:	Paul Michael
Illustrator:	Brian Fox
Research & Editorial Support:	Karen Donnelly-Solomon
Production Manager:	Donna M. Garant

ADVERTISING SALES

Advertising Manager:	Jannine Irizarry
Marketing Assistant:	Melissa J. Bernier

1-508-678-4200
FAX 1-508-675-6002

SPECIAL THANKS TO:
Buddy Terrell & Byrd Press
Bob at Riverside Art, Ltd.
Swansea One Hour Photo
Pride Offset, Warwick, RI
Mach 1 Photo

Amazing Computing™ (**ISSN 0886-9480**) is published monthly by PiM Publications, Inc., Currant Road, P.O. Box 869, Fall River, MA 02722-0869.

Subscriptions in the U.S., 12 issues for **$28.00**; in Canada & Mexico surface, **$36.00**; foreign surface for **$44.00**.

Second-Class Postage paid at Fall River, MA 02722 and additional mailing offices.

POSTMASTER: Send address changes to PiM Publications Inc., P.O. Box 869, Fall River, MA 02722-0869. Printed in the U.S.A.

Copyright© Dec. 1989 by PiM Publications, Inc. All rights reserved.

First Class or Air Mail rates available upon request. PiM Publications, Inc. maintains the right to refuse any advertising.

Pim Publications Inc. is not obligated to return unsolicited materials. All requested returns must be received with a Self Addressed Stamped Mailer.

Send article submissions in both manuscript and disk format with your name, address, telephone, and Social Security Number on each to the Associate Editor. Requests for Author's Guides should be directed to the address listed above.

AMIGA™ is a registered trademark of Commodore-Amiga, Inc.

Amazing Dealers

The following **Amazing Dealers** carry *Amazing Computing*™, your resource for information on the Amiga™, and *AC's Guide To The Commodore Amiga*™, the total Amiga product guide. If you are not an Amazing Dealer, but would like to become one, call.

1-508-678-4200

Alaska
Juneau Electronics — Juneau

Alabama
Abax Data Systems — Huntsville
Alabama Book Store — Tuscaloosa
Madison Book & Computer — Madison
Mel's Photo & Computer Shop — Montgomery
Paeco Industries — Birmingham
The Computer Image — Birmingham
The Computer Nut — Birmingham
Universal Computer System — Mobile

Arkansas
Commodore Connection — Sherwood
Microtronix — Fort Smith
The Micro Shop — Little Rock

Arizona
Bookman's — Tucson
Books Brothers Ltd. — Tucson
Books Etc. — Tempe
Brain Child Software — Phoenix
Computer Library, Inc. — Tempe
Livingston's Books, Inc. — Sierra Vista
North American Digital — Tucson)
Try Tech Inc. — Pheonix

California
3 E Software And Systems — Hayward
Aday International — Foster City
American Mags & Books Ltd. — Englewood
Arcadia Newstand — Arcadia
AugNET — Santa Clara
B. Dalton Books — San Rafael
Balboa Office Supply — San Diego
Bits & Bytes — Fairfield
Bits n Pieces Comp Prod. — Anaheim
Bookland — Marysville
Bookstar — San Diego
Brentano's — Sacramanto
Brentano's — San Francisco
Brown Knows Computing — Redlands
CAL-Tech Bookstore — Pasadena
Candy Computer — Elk Grove
Castro Kiosk — San Francisco
Centerfold News — Los Angeles
Central Park Bookstore — San Mateo
Century Computer Systems — La Habra
Circus of Books — Los Angeles
Circus of Books — Los Angeles
Commodore Land — Alhambra
Commotech — National City
Computer Attic — Palo Alto
Computer Literacy Books — Sunnyvale
Computer Literacy — San Jose
Computer Literacy — Santa Clara
Computer Nook, Inc. — San Bernadino
Computer Outlet — Chula Vista
Computer Outlet — San Marco
Computer Room — Scotts Valley
Computer Showcase — San Francisco
Computer Spectrum — Burlingame
Computer Time — Citrus Hts
Computers Plus — Grove City
Coronet Newsstand — Oceanside
Covina Bookstore — Covina
Creative Computer Systems — El Cajone
Dataphile — Watsonville
Del Amo Books & News — Torrance
Delaurer's News Agency — Oakland
Discount Computer Station Inc. — Long Beach
ECX Computer Co. — Walnut Creek
Empire Computers — Santa Rosa
Encinitas News — Encinitas
Fli-American — San Francisco
Fisher Computers + Software — Rancho Cucamonga
Harding Way News — Stockton
Henrichson, Schneider & Young, Inc. — Berkeley
Home Computing Center — Concord
Home Computing Center — Mill Valley
Home Computing Center — Sunnyvale
HT Electronics — San Diego
Int's Periodical Dist — Santa Ana
International Telecine — Granada Hills
K J Computers — Venice
L. A. Int's News — Placentia
Little Professor Books — Santa Ana
LTC Computers — Berkeley
Lyon Enterprises — Berkeley
Lyon Enterprises — San Jacinto
Micro P. Technologies — Lawndale
NewsBeat — Davis
Norma McCay — San Diego
Northridge Newsstand — Northridge
Op Amp Tech Books — Los Angeles
Paras Newsstand — San Diego
Parsin International — San Diego
Printers Inc. — Palo Alto
Put's Electronics — North Highlands
R&R Software — Chula Vista
Ridgecrest Computer Ctr. Inc. — Ridgecrest
S.O.S. Computers — Los Angeles
Sawyer's News Inc. — Santa Rosa
Software Cellar — Anaheim
Software Central — Pasadena
Software First — Santa Rosa
Software First — Vallejo
Software Plus — Citrus Hieghts
Software Solutions Plus — El Cerito
Stacy's #5 — Modesto
Technitronics Service — South Pasadena
The Booksmith — San Francisco
Third Avenue News — Chula Vista
Tower Books — Chico
Tower Books — Citrus Hts.
Tower Books — Concord
Tower Books — Mountain View
Tower Books — Sacramento
Tower Books — San Mateo
Universal News — Hollywood
Vintage Wine — San Diego
Warren Engineering — San Diego
WCI Computer Central — Eureka
Winners Circle — Berkeley
World Book & News — Hollywood

Colorado
Aurora Newsstand — Aurora
Citadel Computers — Colorado Springs
Computer Discount — Denver
Computer Repeats — Boulder
Computer Room — Aurora
Micro Computer World — Fort Collins
Micro World — Lakewood
Paragon Computers — Colorado Springs
Paragon Software & Electronics — Colorado Springs
Sun Country Computers — Colorado Springs
Tattered Cover Bkstore — Denver
Westerminster Newsland — Westminster

Connecticut
Aetna Life Club Store — Hartford
Amiga Business Computer — Norwalk
Beta Press — Derby
Computer & Software Connections — Plantsville
Dame Electronics — Hamden
Friendly Computers — Plantsville
G & G Christ Church — Derby
Gordon & Gotch — Interconics, Inc. — Woodbury
Int'l Mag Dist/Capetown — Derby
Lac Distributors — Norwalk
Lebanon/Levant Dist. — Derby
Liberty Book — Derby
Micronetics Inc. — Norwalk
One Byte — Quaker Hill
Software City — Orange
Software Kingdom — East Hartford
Tricorn Computers — North Windham

District of Columbia
Programa — Washington
World News — Washington

Delaware
Castle Video — Newark
Newark Newsstand — Newark

Florida
A A Computers — Jacksonville
Ami Comp Computer Center — Oviedo
Bookstop — Clearwater
Bookstop — Miami
Bookstop — Orlando
Brentano's — N. Palm Beach
Chuck's Newsstand — Vera Beach
City Newsstand — Melbourne
Commodore Country Inc. — Pinellas Park
Computer Bar — Pennsacola
Computer Base — Brandon
Computer Image — Miami
Computer Lane Inc — Pembroke Pines
Computer Room — Ft. Lauderdale
Computer Specialties — Melbourne
Computers + Video Creations — Palm Beach
Computers Etc. — Sarasota
Computers Plus — S. Daytona
Creative Equipment — Miami
Education Computers Etc. — Tallahasse
Extra Extra — Tampa
Florida Book Store — Gainesville
Gen-Rod — Miami
Gulf Coast Computer Exchange — PanamaCity
Joe's News, Inc. — Miami
Malvar Freight — Miama
Mark Twain Smoke Shop — Lakeland
Micro Computer Center — Largo
New Age Electronics — Clearwater
New Age Electronics — St. Petersburg
PC Collection — Coral Gabels
Program Store — Clearwater
Program Store — Pinellas Park
Prospect News & Books — Ft. Lauderdale
Public Domain solutions — Sarasota
Software City — Orange Park
Sunny's At Sunrise, Inc. — Sunrise
Sunshine Software — Ft. Lauderdale

Georgia
Ampex System — Norcross
BPJ Enterprises — Roswell
Brentano's — Atlanta
Cardinal Computers — Dalton
Micro Center/Marietta — Marietta
Mostly Software — Columbus
Oxford Books, Inc. — Atlanta
Personal Computer Exchange — Lilburn
Software City Savannah — Savannah
Software South Inc. — Savannah
The 64 Store — Atlanta

Hawaii
Astro Computers & Electronics — Aiea
Computronics — Honolulu
Data One Computer Center — Honolulu
J & C Repair — Aiea

Idaho
A.B.I. Computer & Video — Nampa
The Creative Edge — East Ketchum

Illinois
Audio Visual Req. — Chicago
Books & Bytes, Inc. — Naperville
Computer Software Connection — Moline
Comp—Kingdom — Lisle
Coopersmith — Vernon Hills
Digital World — Naperville
DigitalWorld — Addison
DOS Computer Center — Niles
FD Software & Caddis — Bloomingdale
Keep In Pace Computers — Champaign
Krochi's & Brentano's — Aurora
Krochi's & Brentano's — Chicago
Krochi's & Brentano's — Orland Park
Krochi's & Brentano's — Schaumburg
M Two J Two, Inc. — Champaign
MDJ2 Inc. — Champaign
Memory Expansion — Bloomington
Micro Age — Oaklawn
Micro Ed Enterprises Inc. — Chicago
Micro Tech Soultions — Aurora
Micro Tek — Kankakee
Ring Software — Geneva
Software City — Bellville
Software Plus Chicago — Chicago
Software Plus West — Hanover Park
Software Plus — Wheeling
The Memory Expansion — Bloomington
Unique Computer — Sterling
Warren Radio/FRANCO — Peoria

Indiana
Bookland — Indianopils
Burkat Computer Ctr. — South Bend
Cal Com Inc. — Kokomo
Computer Corner — Ft. Wayne
Computer Corner — Marion
Computer People Inc. — Michigan City
CPU Inc. — Indianapolis
Indiana News — Indianapolis
Micro Computer Centers Inc. — Indianapolis
Programs to Go — Muncie
The Book Corner — Bloomington
The Computer Room of Merrillville — Merrillville
Vons Computers — West Lafayette
Xerox Computer Center — Indianapolis

Iowa
Computer Solutions & Facts — Des Moines
Computers & More — Des Moines
Granert Computers — Des Moine
Soft Search — Ames

Kansas
Art Electronic Distributors — Wichita
Brentanos — Mission
Micro Media — Hays
Micro Computers — Manhattan
Software City Overland — Overland Park
Software City — Overland Park
Software Exhange of KS — Wichita

Kentucky
Computer Dimensions Inc. — Lexington
Computer Mania Inc. — Florence
Expert Services — Florence
Hawley Cooke — Louisville
Hawley Cooke — Louisville
Reality World Computers — Lexington
Soltech Computer Systems — Lexington
The Computer Lab — Louisville

Louisiana
Bookstar — New Orleans
City Newsstand — Baton Rouge
Computer Time — Shreveport
Lakeside News — Metairie
Modern Business Machines — Metairie
Software Center International — Metairie
Software Source — Lafayette
The Book Shelf — Slidell

Maine
Hands On Computing — Westbrook
Skyright Software — Belfast

Maryland
Buried Treasure — Rockville
Capital Classics Ltd — Silver Spring
Greatings and Readings — Towson
Knowledgeware Systems — Kensington
Micro Computer Center — Baltimore
New Age Computers — College Park
Professional Micro Services — Baltimore
Software Advantage — Rockville
Software City — Rockville
Waldorf Computer — Waldorf

Massachusetts
Computer Works — Middletown
Computer Zone — North Attleboro
Dex — Williamstown
HCS Computer Center — Marshfield
LCA Video & Computer Ctr. — Norwood
Memory Location — Wellsley
Newsbreak, Inc. — Swansea
OmniTek Computers — Tewksbury
Pioneer Valley Data Equipment — Amherst
Soft Designs — Watertown
Softpro — Burlington
Software Connections — Swansea
The Bit Bucket — West Newton
The Software Haus — Quincy
Tycom Inc. — Pittsfield

Michigan
Borders Book Shop — Novi
Central Park Computers — Okemos
CompuSoft — Kalamazoo
Computer Emporium — Lansing
Doogsmith's — Livonia
Direct Access — Lansing
Global Computer Center — Dearborn
I/O Software — Flint
Little Professor Books — Brighton
Michigan News Agency — Kalamazoo
Michigan Software — Novi
Micro World — Livonia
Micro World — Milford
Micro World — Roseville
Micro World — Southfield
Micro World — Southgate
Micro World — Sterling Heights
Microworld Computer Center — Dearborn
Midland Computer Shop — Midland
Midwest Micro Computer — Lapeer
New Horizons Bookshop — Roseville
Norman Business Control Sys. — Wyoming

Become An
Amazing Dealer
TODAY!

Readmore Newscenter — Lansing
Rochester Book Center — Rochester Hills
Slipped Disk — Lansing
Slipped Disk — Lavonia
Slipped Disk — Madison Heights
Slipped Disk — New Baltimore
Slipped Disk — Southfield
Slipped Disk — St. Clair Shore
Software City — Grand Rapids
Software House — Kalamazoo
Specialists Computer — Livonia
Sula Street Computers — Ann Arbor
T.J. Enterprises — Clinton
The Edge Connector — Clio
Ye Olde Computer Shoppe — Ypsilanti

Minnesota
Brentano's — Minneapolis
Computer Excercise World — Minneapolis
Computer Stuff — Minneapolis
Computers, etc. — Eagan
JMH Software — Maple Grove
Lock Ahead Software — Northfield
Shindler's — Burnsville
Shindler's — Crystal
Shindler's — Edina
Shindler's — Inver Grove Hgts
Shindler's — Maplewood
Shindler's — Minneapolis
Shindler's — Minnetonka
Shindler's Read More — Minneapolis
Shindler's — Roseville
Shindler's — St. Paul
Shindlers — Saline
Specialists IN — Hopkins
Valiant Inc. — Stillwater
Wizard's Computer Shop — Crystal Lake

Mississippi
National Computer Center — Ocean Springs
Assoc Computer Services — Springfield
MCP Associates — Astoria
MicroWorks — Kenmore
Mnemonics — North Port
QB Software, Inc. — Forest Hills
Readers Market — Tonawanda
Rochester Books — Rochester
Sherry Tunkel — New York
Singapore Book — Huntington Station
Software & Such — Scotia
Software City — Spring Valley
Software House — Fairport
Software Link — White Plains
Software Supermarket II — Cheektowaga
Star Tech Systems — Massena
Syntiare Trading — New York
Sysout Computers — Brooklyn
The Computer Cellar — Albany
The MicroWorks — Tonawanda
Unicom Electronics — Johnson City
Video Computer Bin — Brooklyn
Video Computer Center — Rome
Village Computer Inc. — New York
World Computers Inc. — Hicksville

North Carolina
Digitz — Raleigh
K&S Newsstand — Clemmons
K&S Newsstand — Kernersville
K&S Newsstand — Winston-Salem
News Center — Cary
Parker News Inc. — High Point
Servatronic Technologe — Burlington
Software City — Charlotte
Software City Fayetevl — Fayetteville
Software City — Raleigh
Triad Computers — Winston-Salem

North Dakota
Computer Associates — Fargo
G. F. Computers — Grand Forks
Software House — Fargo
The Computer Store — Minot
Brands Mart Computers — Kansas City
Carthage — Carthage
Carthage Electronics — Kansas City
Clark Worldwide — St. Louis
Data Byte Computer Co. — Independence
Software City — St. Louis
Systems Plus of St. Louis — St. Louis
Systems Plus — Overland Park

Nebraska
My Com Inc. — Plattsmouth
Double E Computer Systems — Omaha

Nevada
Computer House — Sparks
Computer World — Las Vegas
Nat'l Discount Computer — Las Vegas
Tower Records — Las Vegas

New Hampshire
Dartmouth Bookstore — Hanover
Diversified Computers — Keene
System Eyes — Merrimack

New Jersey
Act One — Vineland
Atlantic News — Atlantic City
Business System Professional — Scotch Plains
Connecting Point — Union
Enachpex Enterprises — Fort Lee
Family Computer Centres — South Orange
Garden State News — Cliffside Park
Hudson News — New York
Morris County Stationers — Chester
Program Store — Princeton
Software Station — Rockaway
The Program Store — Eaton Town
The Program Store — Wayne
Unlimited Software Inc. — Rutherford
Village Computer Center — Cedar Knolls
Waldensoftware — Paramus
Wilson Group USA — Jersey City

New Mexico
New Horizon Computer System — Alamogordo
Newsland Bookstore — Albuquerque
Page One Newsland — Albuquerque
Road Runner — Espanola

New York
Amagination — New York
Amcorp Ltd — Huntington Station
Amicus- Hudson Valley — Wappingers Falls
Amiga Business Computer — East North Port
Book Distributor — Huntington Station
Broadway News — Saratoga Springs
Byte Shop — Merrick
Compute Center — White Plains
Computer Metropolis — New Paltz
Computer Outlet — Jamestown
Computers Etc. — Syracuse
Computerware — East Meadow
Contemporary Computing — Latham
Creative Computers — Binghamton
Cromland Inc. — Brooklyn
Global Software — New Hyde Park
Island Software Computer Services — Coran
Kemco Publisher Ser — East Northport
Leigh's Computers — New York
Ohio
BS Software — Columbus
Brentanos — Cincinnati
Cincinnati Fin SQ News — Cincinnati
Computer Network — Newark
Eddy's Cigar Store — Elyra
Fairborn Home Computer — Dayton
Infinity Computing — Columbus
Lakewood International News — Lakewood
Leo's Book & Wine Shop — Toledo
Local Computer Store — Chesterland
Medical Computers — Centerville
Micro Center-Columbus — Columbus
Micro Center-Westerville — Westerville
Microtech Computer Center Inc. — Toledo
Microware Magic — Fairfield
News Shop — Kent
North Coast Programming — Willoughby
Not Just Software — South Toledo
Not Just Software — West Toledo
Paser & Ink — Tiffin
PC Powerhouse, The — Fairfield
Power Computing Inc. — Toledo
Salem Computer Center — Salem
Software Centre International — North Olmsted
Software and More — Cincinnati
Software Connection — Mayfield Hts.
Somers Photo & Comp. — Cleveland Heights
Logrods — North Olmstead

Oklahoma
Little Professor Books — Stilwater
Merit Computers — Oaklahoma City
Micro Link Computer Store — Tulsa
Second Hand Software — Oklahoma City
Video-Comp Inc. — Lawton

Oregon
Andromeda — LaGrande
Clackamas Computers — Beaverton
Clackamas Computers — Clackamas
Comm-Shack — Salem
Libra Books Inc. — Eugene
News & Smoke Shop MF — Medford
PC Express — Eugene
Powell's Tech — Portland
Rich Cigar Store — Portland
Sixth & Washington News — Portland
Soft Cell Software, Inc. — Portland
Software Pipeline-Salem — Salem
Software Popline — Beaverton
Tech Books — Portland

Pennsylvania
Alpha Omega EBE — Warren
ATD Software — Sayre
Basic Computer Systems — Erie
Computer Basics Inc. — Hermitage
Bryn Mawk News — Bryn Mawk
Circle Computer — Ephrata
Clark Worldwide — Middletown
Computer Wurx — Stroudsburg
Computers and Games — Reading
Downington Computer Center — Downington
Elmo's Computers — Brownsville
Floppy House Software — Palmyra
Gene's Books — King of Prussia
Graphix Computer Center — Lionville
Lauer-Todd Computer Solutions — Newtown
Micro Master Inc. — Monaca
North Central Digital Systems — Danville
Pittsburgh Computer Store — Pittsburgh
Software Hut Inc. — Philadelphia
Some Hole In The Wall — Philadelphia
T A Slick & Co, Inc — Lancaster
Tasco Computer Center — Lancaster
The Floppy Disk — Mechanicsburg
Triangle Computers — Indiana
Walden Hill Images — Berlin

Rhode Island
Micro Limits — Smithfield
Software Connections — Warwick
The Computer Place — Woonsocket

South Carolina
Avcom International Inc. — West Columbia
Software Haus — Charleston Hts
Watson's Jewelry — Chesterfield

Tennessee
Bristol Micro Technolog — Bristol
ComputerLab of Memphis — Memphis
Computer Hardware Services — Nashville
T.Y. Corp — Bartlett

Texas
A&E Computers — Grove
All Systems Go Software — San Antonio
Bookstop — Arlington
Bookstop — Austin
Bookstop — Dallas
Bookstop — Houston
Bookstop — San Antonio
Brentano's — Dalias
Byte Shop — Austin
Computer Ease — Corpus Christi
Computer Magic — Austin
Computer Skills — Eules
RNS Computer Services — Texarkana
Software Plus — Desota
Metropolitan Computer — Richardson
Micro Search — Houston
Software Library — Wichita Falls
Software Terminal — Fort Worth
The Computer Experience — San Antonio
The Computer Stop! — Amarillo
The Federated Group — Mosquite
Trinity News Company — Fort Worth

Utah
Armadillo Bros — Salt Lake City
Magazines and Gifts — Salt Lake City
People's Computers — Provo

Vermont
Ray Supply Inc. — South Burlington

Virginia
Audio Fidelity Corp — Richmond
Brentanos — Alexandria
Discover — Fairfax
Family Computer Center — Fairfax
Family Video & Computer Center — Newport News
I.Q Computer — Norfolk
Tinact Software — Prince William
Video Computer Retailers — Chesapeak
Virginia Micro Systems — Woodbridge

Washington
A.P.P.L.E. CO-OP — Renton
Amazing Electronics — Spokane
Apple Valley Computers — E. Wenatchee
Bits, Bytes and Nibbles — Spokane
Brentano's — Seattle
Bull Dog News — Seattle
Byte World — Kent
Computer Quest — Edmonds
Computers + Inc. — Tacoma
Family Computers — Everett
Micro Age Computer Store — Bellevue
Nybbles & Bytes — Tacoma
Omni International — Seattle
Port Book and News — Port Angeles
Steve's Broadway News — Seattle
Swan Computers — Yakima
Take A Byte — Kennewick
Tech Star — Kent
The Computer Shoppe — Vancouver
Tower Books — Bellevue
Tower Books — Seattle
Tower Books — Tacoma
Y Not Magazines and Books — Federal Way

Wisconsin
20th Century Books — Madison
Computer Software Center — Milwaukee
Fox Valley Personal Computer — Oshkosh
Madison Programmer's Guide — Madison
Mom's Computers — Stevens Point
Software City — Racine
TMW Software Inc. — Wausau

Wyoming
Byte Emporium — Laramie
Computer Works — Laramie
Crazy Mountain Elect — Casper

Outside of the United States

Canada

Alberta
Coles, Book People — Calgary
Coles, Book People — Edmonton
Software Supermart — Edmonton,

British Columbia
Computime Computer & Software — Prince George
Strider Computer Centres — Delta
Unihall — Vancouver

Manitoba
Advanta Software, Ltd. — Winnepeg

New Brunswick
Computer World — Moncton
P.G. Compusystems Inc. — Campbellton

New Foundland
J.W. Allan Co. Ltd. — St. John's

Nova Scotia
Atlantic News — Halifax
Coles, Book People — Yarmouth
Kobetek Systems — Halifax
Renmat Enter Ltd. — Greenwood

Ontario
All Computers — Gloucester
AKon Electronics Ltd. — Toronto
Brittons Smoke Shop — Ottawa
Canadian Software Source — Downsview
Chatham Computer Sales — Chatham
Classic Computer — London
Coffey's News Depot — St. Thomas
Coles Bookstore — Rexdale
Coles, Book People — Guelph
Coles, Book People — Scarborough
Coles, Book People — Toronto
Coles, Book People — Willowdale
Compul-or Software — Barrie
Computer & You, The — Islington
Computer Express — Mississauga
Computer Kingdom — Sudbury
Computer House — Chatham
Computer Odyssey, Inc. — Hamilton
ComputerLand — Cornwall
Computronics — Brampton
Computronix Data Centre — Toronto
Comspec — Toronto
Con-pute — Oshawa
Cryston Computer System — Midland
CVF Data Systems — Peterborough
Daily Planet, The — Whitby
Data Marketing Kingston — Kingston
Desktop Computer, Inc. — Kitchener
Electronics 2001 — Willowdale
Foreign Correspondence — Kingston
Globe Magazine — Ottawa
Gran Bookstore, The — Toronto
Granger T.V. Ltd. — Goderich
Hi-Land News — Barrie
Horizon Computers — Whitby
Humber College Stores — Rexdale
Kahle Electronics — Kanata
Lichtman's — Thornhill
Lichtman's — Willowdale
Lightman's News & Books — Toronto
Litchtman's — Toronto
Lynx Technical Services — Ottawa
Mags & Fags — Ottawa
Mikadon Computer Engineering — Thunder Bay
Mr. Diskettes — Ottawa
Novatech — Niagra Falls
Nu-Info Teck — Ottawa
Peach Micro — Ottawa
Personal Computers — London
Personal Keyboard Systems — London
Prospero Books Ltd. — Ottawa
Smoker's Corner — Pickering
Smoker's Place — Oshawa
Smokers Plus — St. Catherines
The Software Project — Hamilton
U of W Bookstore — Waterloo
World's Biggest 200 — Toronto
Zoez Software — Brantford
Zoez Software — Milton

Quebec
Azimut Infor. Inc. — Quebec
Camelot-Info — Montreal
E. M. Press & Books, Inc. — Montreal,
Jupiter Technologies Inc. — Fi-Claire
Logicel International — Montreal
Multimicro — Ville D'Anjou
Ordinateurs 2000 Inc. — Hull
Quebits Logicel Inc. — Montreal
Star Tek Inisolutions — St. Leonard

Saskatchewan
CD's Computer Connection — Saskatoon,

Distributed in Canada by
Ingram Software Distributors — Concord
Compulit Distributors — Britain Columbia

Argentina
Logic Computer — Buenos Aires

Australia
Computermart — Leederville,WA
Diskworks — Greenwich, NSW
Electronic Centre — Cloverdale, WA
John Vertrowen — Abbotsford, Melburn
Maxwell Office Equipment — Melbourne
Run Soft — Melbourne

Saudia Arabia
Abdulla Fouad & Sons — Daman

Scotland
The Amiga Center of Scotland — Edinborough

Switzerland
John Vertrowen — Cloverdale, WA
Maxwell Office Equipment — Abbotsford, Melburn
Run Soft — Melbourne

Saudia Arabia
Abdulla Fouad & Sons — Daman

Scotland
The Amiga Center of Scotland — Edinborough

Sweden
Nicholas Pearson — Lund
Procomp — Mohed

Amazing Computing™ is also available in most B. Dalton Booksellers, B. Dalton Software Stores, Crown Books, Software Etc., selected WaldenBooks Stores, and Walden's Software Store locations.

Feedback Forum

DEAR AC:

Since Commodore is putting a large effort into advertising the world's best home computer let us all help make 1990 the " Year of the Amiga". Are you tired of the MS-DOS talk at work? While your Amiga sits at home with its superior operating system that multitasks and even comes with a graphics co-processor. You can also add an MS-DOS clone inside your Amiga box if you own an A2000 or A2500.

If each Amiga owner can convince an employer or a friend to purchase this great machine. We could protect our investment and double its ownership to two million.

Have a video slide show at work using your Amiga. I did it and now everyone in my office knows what an Amiga is. They are even asking where they can get software for their MS-DOS machines (chuckle). The software I used was TV*TEXT, TV*SHOW, and the Director. I had some shareware music in the background. for narration I used a mixer and microphone I had purchased. Give it a try— your office needs an Amiga linked to a good VCR for developing presentations!

If your office only has PC Compatibles, bridgeboard is for them it is a 100% compatible PC Clone on a card. there is also an emulator (the Transformer) that's a useful tool for some MS-DOS programs. The Transformer will run an MS-DOS D410 emulator for the Data General Mini Computer at full speed Amiga.

Be sure to let them know that wotj A-Max the Amiga can run many Mac programs. This is a combination of software and hardware and is not the same (or as good) as the bridgeboard clone inside your Amiga box. Tell them that the Amiga is a very powerful computer with the same CPU as a Mac plus it has graphics and sound co-processors. If MS-DOS compatibility is a must, then the Amiga is the best choice neither the Mac nor IBM can emulate the Amiga (not powerful enough).

Terminal emulation is very important in an office that has to talk to Mini Computers or Mainframes. There are several excellent Amiga terminal emulation programs available. One of the best Amiga terminal programs currently available is in the Public Domain on Fred Fish #226 and is called VLT. Is your mainframe or mini at work linked with Textronix terminals? If so, you can generate on screen graphics with your Amiga (4105 emulation).

In business software (non-graphics word processors) the Amiga has Excellence. Excellence is much more sophisticated than Word Perfect , but we need the latter so that MS-DOS users can make the transition to the Amiga. There are some excellent spreadsheets available for the Amiga which don't even compare to anything MS-DOS has so I won't mention any by name. Data bases are another story. Superbase Professional is an excellent product, although it is missing oracle since it has become a universal standard.

Have you noticed the new status symbol in the office? The $10,000 laptop with 386 processor. Lets hope that Commodore does indeed come out with the A3000 this year! Power impresses the business world. Lets also hope that when these two are compared that Commodore can promote some unique benchmark tests that will show off the graphics co-processor in the Amiga. I hope that the 386 bridgeboard is available before the 486 becomes the fad.

We have heard that Commodore's implementation of UNIX is very good. If they offer an A3000 UNIX it may be a way of getting a foot in the door of the business world. It is my gut feeling that UNIX will be the next major operating system for the office.

Since we have sold an Amiga to the office, what about the 13 year old next door who is getting bored with his Nintendo? Get at least one outstanding game for your Amiga (even if you hate games) and show the kid what a real computer can do. Some people have bought MS-DOS machines to play games—isn't that disgusting.

Commodore can do it with our help—make the Amiga known to America—it's a great machine, so lets help out. One of the most exciting things about owning an Amiga is the loyalty of the users. Have you ever checked out the public domain programs available for the Amiga—it's great! That's because creative folks know a great machine when they see one and want others to know too. So, let us all make 1990, "The Year of the Amiga"!

Dan Greene
Boulder, CO

-Your enthusiasm is contagious. Lets hope that more people catch it. We agree with your comment on PD software. The people who write PD software are doing the Amiga Community a great service and we thank them. -Ed

Dear AC:

Thank you for Publishing my article "Multitasking in Fortran." I think that the article will provide the means for a substantial number of people to deal with a significant Problem on the Amiga in the Fortran environment.

Unfortunately, an error into the article during the editorial process that must be addressed. In the copy that I submitted to you, my last sentence of text stated; "Frankly, I consider Fortran to be easier to program in than C, for those things that Fortran is good at." It was edited to say; Frankly, for the features that it has, I find Fortran easier to program in than C."

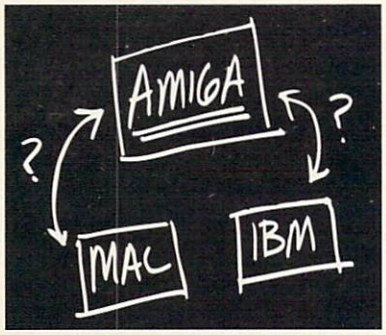
The latter statement is incorrect, in that it implies that I consider Fortran to have more or better features than C. Implicitly, it states that I consider Fortran to be superior to C. I do not consider Fortran to be a better language and I would not start a new project in Fortran, unless there was a compelling reason to do so. Relative to most of the later generation languages, Fortran has some significant limitations. However, it is easier than C to program, for those things that it can do.

I consider Fortran to be valuable mainly because of the huge body of software that already exists, especially in the scientific arena, that is written in Fortran. Accessing that wealth of proven software, while taking advantage of the Amiga environment, was the purpose and motivation for the article.

Sincerely,
Jim Locker
Fairborn, OH

-We appologize for the mistake and hope that it didn't cause any problems. Thanks for the corrections Jim. -Ed

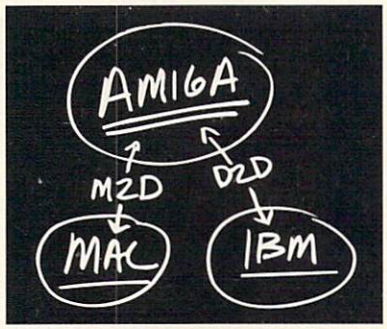
•AC•



PROBLEM

Your boss wants you to produce an animated video presentation to help sell that new project to top management. You know that a full-color, animated presentation, developed on the Amiga, using its powerful desktop video software, will make a convincing impact.

The Amiga is the right machine for the job, but how can you easily import the images and data you need for your presentation from other divisions of the company, data and images which come from Macintoshes and IBM PCs?



SOLUTION

File transfer programs MAC-2-DOS and DOS-2-DOS from Central Coast Software! Using these simple and easy-to-use Amiga programs, you can now quickly and easily transfer the Mac/IBM/Atari data and images you need to and from the Amiga.

MAC-2-DOS connects a Mac floppy drive directly to the Amiga • Reads and writes 400K/800K Mac disks • Converts MacPaint images to/from IFF • Imports Mac clip art for use on the Amiga • Converts ASCII text files both ways • Converts PostScript files both ways • Supports MacBinary format • Includes conversion utilities for PICT files and Mac fonts • Creates icons, as necessary • Formats 400K/800K Mac disks.

DOS-2-DOS uses your Amiga's floppy drives to read/write IBM/Atari 3.5-inch disks • Reads/writes 5.25-inch IBM disks (using an external 5.25-inch Amiga drive) • Converts ASCII text files both ways • Supports international character sets • Formats IBM/Atari disks • Use with IBM program HIJAAK (from Inset Systems, not included) to capture/convert IBM graphic images to/from IFF.

MAC-2-DOS* — *when it absolutely, positively has to get to or from a Mac disk, immediately.

DOS-2-DOS* — *when it absolutely, positively has to get to or from an IBM PC (or Atari) disk, immediately.

And to preserve your entire presentation, all supporting data and graphic files, as well as the programs:

QUARTERBACK* — *the fastest and most reliable Amiga hard-disk backup program!

CCS

Central Coast Software™

424 Vista Avenue Golden, Colorado 80401
(303) 526-1030 • (303) 526-0520 (fax)
Dealer Inquiries Welcome

MasterCard VISA

Circle 145 on Reader Service card.

Guest Commentary

More Advice For A1000 Upgrades

Richard Taylor, President of Expansion Technologies Speaks Out.

I want to thank Amazing Computing for the opportunity to respond to some of the points made by Mr. Saunders in his November article "Learning the Hard Way, The frustrations of upgrading your A1000" (Volume 4 Issue 11).

It's unfortunate that one persons experience is presented as an article and the corrections or explanations are delegated to the editors page. The article left a very negative feeling for A1000 owner wishing to expand their computer. In a marketplace that has largely abandoned the A1000 we have another unnecessary and largely undeserved nail in the coffin.

As one of the original Amiga hardware manufacturers we have been answering our customers A1000 questions for a long time. We are still learning the variables of the A1000 and yes it has at times been very frustrating. Some of the rules are learned as we go along. The newer rules apply to the use of A2000 cards on the A500 and A1000 bus.

We all know that the A1000 expansion bus was not created perfectly. It has specific limits and will not do everything for everyone. However, it does do some things very well if certain rules are followed. If the A1000 user wishes to expand beyond those safe parameters it is suggested they evaluate moving up to an A2000 now and not later.

The Tool Box for the A1000 was created for those who may upgrade to an A2000 and want an economical and reliable way to add memory and a hard drive. If these Are A2000 upward compatible products then they do not have to repurchase this capability. But the limits must be observed.

In the case of Mr. Saunders there are many people in the field working with his specific combination of products. There are definite reasons why he was not as successful as others and they are covered later. The troubles he experienced were very real and the lessons he imparted should not be overlooked. The point being made is many have successfully expanded their A1000 with out it being a nightmare. With either a Tool Box or another manufacturers expansion product you should not be hesitant to try. But you must be careful and take the recommended precautions.

In a non technical manner I would like to point out some of the precautions to be taken when expanding your A1000 with A2000 cards. Then we will cover some of the troubles mentioned in the article.

1. SOME A2000 CARDS WORK BETTER THAN OTHERS ON AN A1000. It has to do with the differences in the timing of the A1000 bus as opposed to the A2000. The A500 bus is closer to the timing of the A2000 bus. Your chances of success are much higher almost to the point of it not being an issue. Whether a card works or not varies and sometimes depends on the tolerances of the machine. Because of this it is best to go with what works for most people in the field. Use only recommended cards that have proven themselves to work. Call us directly (415656-2890) or ask your dealer to for the latest information before you purchase your card.

In defense of the manufacturers whose cards do not work on the A1000 it must be remembered that working on the A500/1000 bus was not an engineering requirement. The chances are a card cage like the Tool Box did not exist when they were defining the performance parameters of their product.

2. GROUND AND UPGRADE THE PALS. The A1000 bus will reliability accept two devices if you ground and upgrade (if required) the PAL chips on the inside of the machine. An expansion product plugged into the 68000 socket, like the Insider memory board, is also considered a device on the bus. Many articles have been written on this problem including some in this magazine so I will not go into detail. The problem and solution are covered fully in the Tool Box manual. The PAL chips can be purchased from several sources including Expansion Technologies. A reprint of the instructions is also available without charge by calling our office. Armed with this information any service center and/or most customers can make the modifications.

As one of the original Amiga hardware manufacturers we have been answering our customers A1000 questions for a long time. We are still learning the variables of the A1000 and yes it has at times been very frustrating

3. KEEP YOUR AMIGA AS STANDARD AS POSSIBLE. I totally agree with Phil that trouble shooting a system with nonstandard expansion products is nearly impossible. If you experience any trouble remove all products and try and isolate the problem by using then one at a time. If they work individually and not together it generally means the PAL/GROUND problem. Also let the sales or service person know the combination of products you have or wish to purchase. This is so they can evaluate the total system.

Let's address Phil Saunders problems directly.

1. His first problem was in using a GVP interface. This is a very excellent product but is one that will not work reliably on the A1000. It was recommended that he use a FlashCard with his Quantum.

2. The FLASHCARD/QUANTUM worked fine as long as he didn't try and use KickBench 1.3. This is because both the Flash Card and the KickBench were trying to load WorkBench. An "nonstandard" situation.

3. He experienced the PAL/GROUND problems and unfortunately received bad replacement PALS causing a great deal of frustration. After replacing the PALS the article did not indicate that they were ever grounded. This would explain why he would only work with one device on the bus.

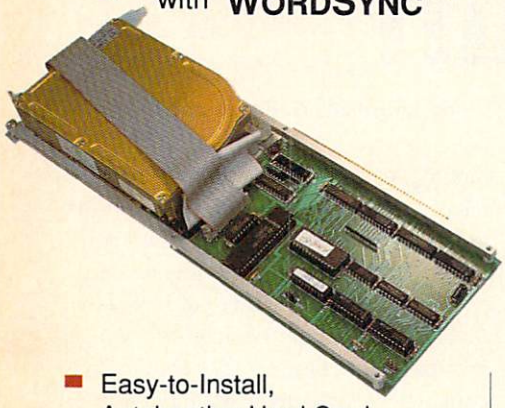
There are a lot of us who still own, and love our A1000 Amigas. If caution is taken they can be reliably expanded to increase their life and usefulness. The problems that Phil Saunders experienced are the exception and not the rule.

Richard Taylor, President -Expansion Technologies

-Amazing Computing believes that there is always more than one side to any story and AC will always be available to both sides of an issue. We thank Mr. Taylor for his advice and comments. -Ed

THE BEST FOR YOUR AMIGA® 2000

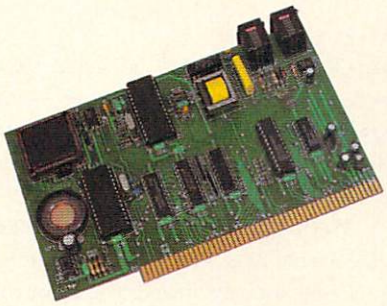
SupraDrive™ with WORDSYNC™



- Easy-to-Install, Autobooting Hard Card with WordSync™ Interface
- WordSync Interface Kit also available separately
- Interface transfers 16 bits at once, giving DMA Speed Without DMA Problems
- Eliminates the sound, video, and serial I/O interruptions found with DMA iinterfaces
- 1-slot design & SCSI-out port
- Easy On/Off Autoboot Jumper
- Compatible with Amiga Bridgeboard, RAM, digitizers, and other boards
- Includes SupraBoot & SupraTools — two full disks of utility software
- Up to 30 Partitions
- All available file systems supported: FFS, MS-DOS, Unix, Macintosh, etc.
- Access to low-level parameters without maintaining MountList
- Excellent Technical Support
- Made in the U.S.A.

30MB (40 ms.)	**$649**
40MB (11 ms. Quantum)	**$799**
80MB (11 ms. Quantum)	**$1099**
40MB SyQuest Removable	**$1199**
WordSync Interface Kit	**$199.95**

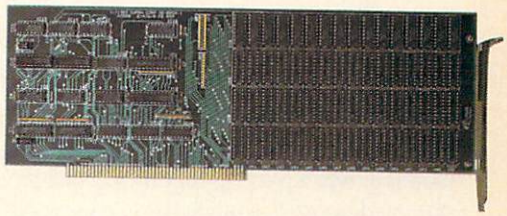
SupraModem 2400zi™



- Half card modem fits in any Amiga bus slot
- Up to 5 modems per computer
- Works with all popular Amiga telecommunications software
- 100% compatible with the industry-standard "**AT**" command set
- Asynchronous 2400/1200/300 baud operation
- Compatible with Bell 103/212A and CCITT V.21/V.22/V.22bis
- Autoanswer/Autodial (tone or pulse)
- Adjustable-volume speaker
- Nonvolatile memory stores custom modem configuration and one telephone number
- Includes free subscriptions to popular on-line services
- Made in the U.S.A.

2400zi	**$179.95**

SupraRam 2000™



- 2, 4, 6, & 8MB configurations available
- Installs easily into any Amiga internal expansion slot
- Easy to expand from your initial configuration — Start with 2MB & add RAM at your convenience
- 6MB configuration allows for maximum benefit with the Amiga Bridgeboard
- Lets you run larger and more sophisticated programs
- Allows creation of large and extremely fast RAM disks
- Test mode & test software make troubleshooting easy
- Made in the U.S.A.

8MB RAM Board 0MB	**$199**
with **2MB** Installed	**$449**
with **4MB** Installed	**$649**
with **6MB** Installed	**$849**
with **8MB** Installed	**$1049**

AVAILABLE AT YOUR LOCAL DEALER, OR CALL:

Supra Corporation

1133 Commercial Way Albany, OR 97321
Voice: 503-967-9075 Fax: 503-926-9370

ORDERS: 1-800-727-8772

Circle 168 on Reader Service card.

SupraDrive, WordSync, SupraModem 2400zi, and SupraRAM 2000 are trademarks of Supra Corporation.
Amiga is a registered trademark of Commodore-Amiga, Inc.

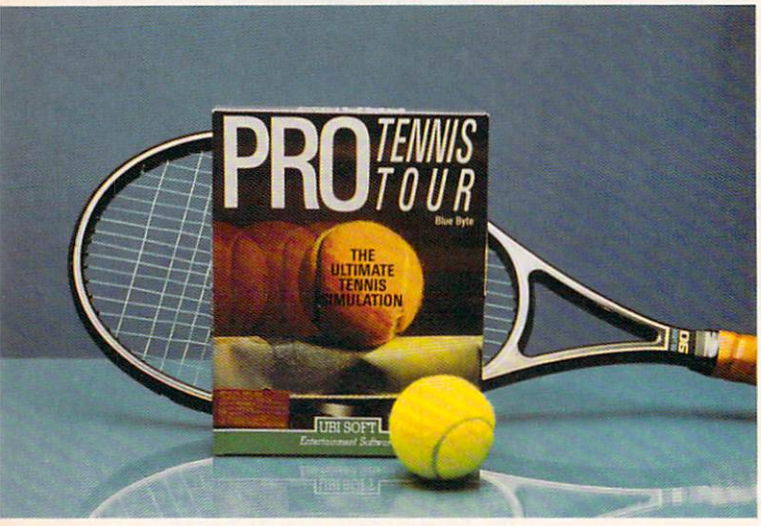
New Products and Other Neat Stuff

by Elizabeth G. Fedorzyn

CREATING QUITE A RACKET

Ever wish you could go forehand to forehand against Ivan Lendl at the U.S. Open, or wiz a passing shot by Martina at Wimbledon? Well, it might be time for you to wake up and smell the Gatorade, as the chances of your encountering such tennis greats on the court are unlikely. Even if you did, your chances of surviving, say, a full-force Lendl serve are probably dismal at best. Of course, if you did get a bit of practice in, maybe a few matches — you know, sort of work up to that day of confrontation — who knows? Pro Tennis Tour might be a good place to start.

Produced by UBI Soft and distributed by Electronic Arts, **Pro Tennis Tour** is a tennis simulation that places you in the role of a professional tennis player seeded 64th in the world. In order to improve your ranking, you must participate against players of varying ranks in such major tournaments as the Australian Open, the French Open, The U.S. Open, and Wimbledon.



A real racket: Pro Tennis Tour by Electronic Arts.

Although you don't get to become involved in lengthy bellicose exchanges with the judges or sign major endorsement contracts, you do get to watch your most impressive shots in 3-D instant replay, as well as practice your forehand, backhand and serves with six practice programs containing three levels of difficulty. And each tournament is played on the same surface as the real games (e.g., grass courts at Wimbledon).

Pro Tennis Tour allows one player to compete against the computer or two players to compete against each other. The game sports sound effects as well as impressive graphics that include an on-screen scoring display, line judges who watch every move you make, and opponents who leap when you place a shot to their upper body area — just like in real life!

Pro Tennis Tour
Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404
(415) 571-7171
Price: $39.95
Inquiry #227



Found in space: Distant Suns by Virtual Reality Laboratories.

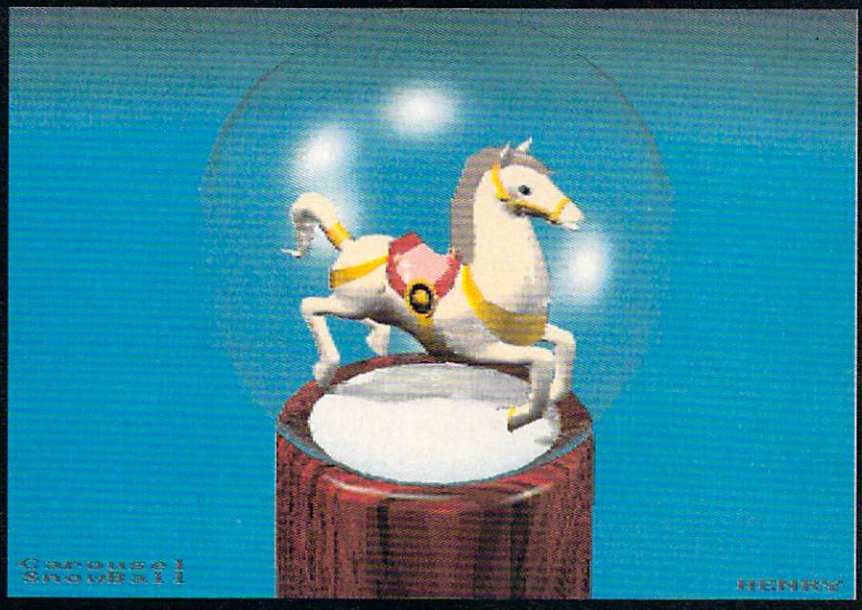
BILLIONS AND BILLIONS...

How many heavenly bodies are there in the solor system? Well, as Steve Martin facetiously pointed out in *Roxanne*, we know it's more than fifty. **Distant Suns** might be able to give you a better idea.

Designed by astrophysicist Mike Smithwick and originally released as Galileo, this Amiga astronomy program has undergone some improvements and is now being produced and distributed as Distant Suns by Virtual Reality Laboratory.

(continued)

AmiEXPO Presents
THE 1990 AMIGA ART CALENDAR



A Complete 1990 Calendar • 13 Pieces of Original Amiga Art
Produced Exclusively On The Amiga
Each work 9" x 13" • Suitable for Framing

ORDER NOW AND SAVE $5.00
The 1990 Amiga Art Calendar is only $14.95
Coupon Worth $5 Savings On Any AmiEXPO

To place an order for your 1990 Amiga Art Calendar, either complete the included form and return to our offices or call us at 1-800-32-AMIGA. To place a telephone order, you must have a valid Visa or MasterCard. Please allow 6-8 weeks for delivery.

ORDER TODAY!

Order by Mail or Call 800-32-AMIGA Nationwide (or 212-867-4663)
For Your All Amiga Art Calendar!

❑*Yes, I want my 1990 Amiga Art Calendar*

_____ *Calendars @ $14.95* _____
Shipping &Handling
Please add $2 per calendar _____
TOTAL _____

Make Check or Money Order Payable to:
AmiEXPO 211 E. 43rd St., Suite 301
New York, NY 10017

NAME _____
COMPANY _____
ADDRESS _____

CITY _____ STATE _____ ZIP _____

For ___ MasterCard or ___VISA Payment
Expiration Date _____
Account Number _____
Name as it appears on card: _____
Signature _____

AmiEXPO™ is a registered trademark of AmiEXPO, Inc. © 1989 AmiEXPO, Inc.

Circle 115 on Reader Service card.

Distant Suns features realistic color with full-screen sky viewing option, 2,200 stars, expandable to 9100, and 450 deep sky objects — all in an astronomically correct sky representation. The program lets you view the sky from any place on Earth at any time from 8,000 B.C. to 12,000 A.D. Why, if Carl Sagan had had access to Distant Suns, he probably could have dragged out his celebrity status a few more light years.

Distant Suns will just make it on 512K memory. Expanded memory is strongly suggested. The package includes a 100-page Users Manual to help you steer through space. Users of the original Galileo program may upgrade to Distant Suns by contacting Mr. Smithwick at 25215 La Loma Drive, Los Altos Hills, CA 94022.

Distant Suns
Virtual Reality Laboratories, Inc.
2341 Ganador Court
San Luis Obispo, CA 93401
(805) 545-8515
Price: $69.95
Inquiry #228

CRIME THAT PAYS

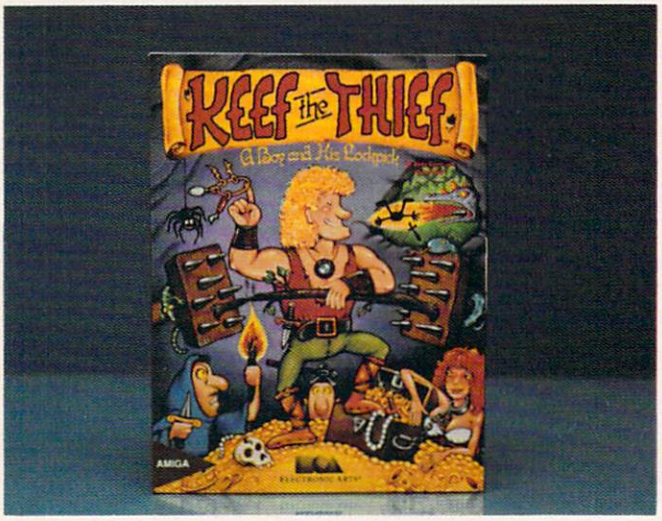
Meet Keef. Keef steals. Keef sells stolen goods. Keef is never without his lockpick. OK, so maybe he's not Ward and June's Dream Child, but he may just be the next Teenage God-King.

Designed by Naughty Dog, Inc. and released be Electronic Arts, **Keef the Thief** is a role-playing game that puts you in the mindset of Keef, a juvenile delinquant who has been banished from his hometown to the Tri-City Area, a vile jungle teeming with slithering creatures and not many good restaurants. As Keef, it is your goal to become absolute ruler of everything and everyone. This is no small task, considering the nasty conditions under which you must now survive and the fact that an evil magician is hard at work trying to create the magic necessary to control the continent.

But you are not without your own advantages. As Keef, you have no problem with the idea of breaking and entering, or stealing supplies, of trying to sell stolen goods — in effect, no morals to hinder you from realizing your goal. (Look out Maury Povich!)

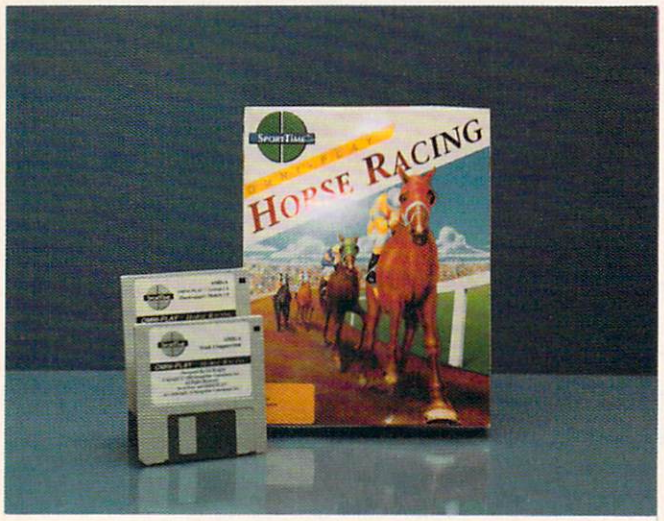
The game features over 200 species of nasty monsters, more than 50 color screen illustrations, 25 complete levels of dungeon, and three complete cities to loot to your delinquant heart's content — not to mention a quite humorous, refreshing approach to text-adventure authoring.

Keef the Thief requires a minimum 512K memory, and can be controlled using either the keyboard or the mouse. Keef incorporates an icon-based interface, and allows a Save Game option to allow you to maintain your current level of debauchery.



A boy and his lockpick: Keef the Thief from Electronic Arts.

Keef The Thief
Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404
(415) 571-7171
Price: $49.95
Inquiry #229



A day at the track: OMNI-PLAY Horse Racing from SportTime Software.

GOING TO GREAT LENGTHS

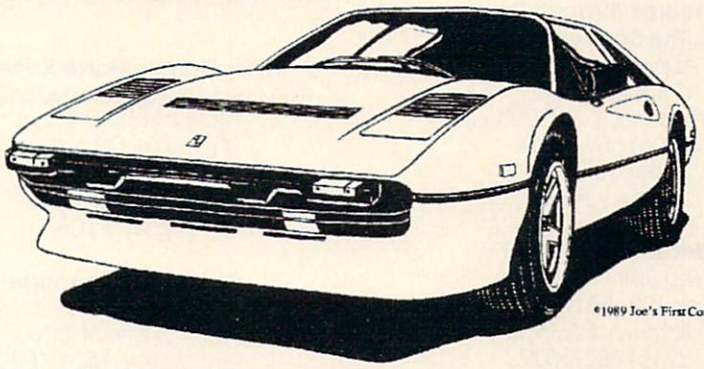
Psst! Disco Dancer in the fifth! That's the only tip I'm gonna give you. For any more, you'll have to head down to the track, or in this case tracks. From the folks who brought you expandable basketball comes a new expandable horse-racing program.

New from SportTime Computer Software, **OMNI-PLAY Horse Racing** lets up to four players bet against each other and up to 19 computer opponents on 128 different horses. If you're the more reserved type, bet on the projected favorites to Win, Place, or Show. But if you're from the Oscar Madison school of pony picking, you might want to be a bit more adventuresome and go for the long shots.

(continued)

Pictures Speak Louder Than Words...





©1989 Joe's First Company Inc.

PIC-MAGIC™

Professional Clip Art Series
Package One

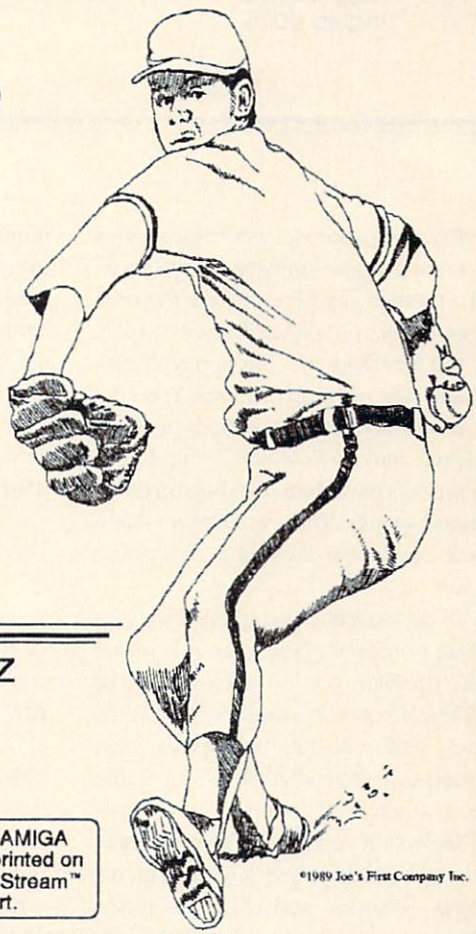
©1989 Joe's First Company Inc.

$99.95 US DOLLARS
(INCLUDING SHIPPING VIA UPS!)

PACKAGE ONE: General Images.

TEN DISK SET * COMPLIMENTARY FAN-FILE™
*** 220 Page Manual (Full Printouts of every image) ***
IFF FILE FORMAT. (Over Screen Sized Bit-Maps)
Over 250 Images from Food to Sports to Cars!

CALL ABOUT AND LOOK FOR:
*** PACKAGE TWO (Release: JAN 1990)**
***** POSTSCRIPT™ (Vector Graphic) VERSIONS**
******* PACKAGE ONE SINGLE DISK SAMPLERS**



©1989 Joe's First Company Inc.

To ORDER or for INFO (In USA) Call:

1-800-387-8967

Joe's First Company

P.O. Box 579, Station Z
Toronto, Ontario
M5N 2Z6 Canada
Tel (416) 322-6119
Fax (416)489-1620

This Ad was created in PageStream™, on an AMIGA 2500 (512k Chip, 2.5 Meg. Fast Ram). It was printed on an HP DESKJET In 300 DPI mode using PageStream™ Fonts Disks and only Pic-Magic™ Clip Art.

JOE'S FIRST COMPANY INC.

Circle 178 on Reader Service card.

OTHER PRODUCTS RECEIVED

Adventures Through Time—Vol. I: The Scavenger Hunt
Aurum Software
P.O. Box 5392
Ventura, CA 93003
(805) 659-3570
Price: $49.95
Inquiry #233

Bride of the Robot
Free Spirit Software
P.O. Box 128, 58 Noble Street
Kutztown, PA 19530
(215) 683-5609
Price: $39.95
Inquiry #234

Colossus Chess X
Artworx Software Company, Inc.
1844 Penfield Road
Penfield, NY 14526
(716) 385-6120
Price: $34.95
Inquiry #235

Daily Double Horse Racing
Artworx Software Company, Inc.
1844 Penfield Road
Penfield, NY 14526
(716) 385-6120
Price: $29.95
Inquiry #236

Dr. Doom's Revenge
MicroProse
180 Lakefront Drive
Hunt Valley, MD 21030
(301) 771-1151
Price: $39.95
Inquiry #237

Joan of Arc: Siege and the Sword
Broderbund Software, Inc.
17 Paul Drive
San Raphael, CA 94903-2101
(800) 521-6263
Price: $44.95
Inquiry #238

Mail-O-Dex
KarmaSoft
P.O. Box 1034
Golden, CO 80402
(303) 277-1241
Price: $49.95
Inquiry #239

Quarterback
Central Coast Software
424 Vista Avenue
Golden, CO 80401
(303) 526-1030
Price: $69.95
Inquiry #240

Tank Attack
Artworx Software Company, Inc.
1844 Penfield Road
Penfield, NY 14526
(716) 385-6120
Price: $29.95
Inquiry #241

Whatever approach you choose, you will have the opportunity to be well-prepared. Statistics available include the past ten race results on all 128 horses at each track, the Morning-line Odds, payoff displays, and post position histories. You also have access to complete jockey profiles. And leave Randy Newman behind, because jockeys have their own personalities and mood swings. There are also a variety of track conditions including dirt, grass, and mud.

Of course, should you tire of playing the same ponies or playing the role of the hapless gambler, you can pick up one of SportTime's expansion modules like Track 2 (a new park with 128 new horses, each with their own individual skills), or Stable Owners (own and operate your own stable, selecting your horses and jockeys).

OMNI-PLAY Horse Racing features digitized sounds, and high-resolution graphics that include dual scrolling background and track. The package requires a minimum 512K memory and consists of a two-disk set (Track 1 and the Handicappers Module 1.0). Future option modules include Jockey Competition, Fantasy Betting, and Prediction Analysis.

OMNI-PLAY Horse Racing
SportTime Computer Software
Distributed by Broderbund Software, Inc.
17 Paul Drive
San Raphael, CA 94903-2101
(800) 521-6263
Price: $49.95
Inquiry # 230

BIG TIME

Well, if your Amiga has appeared a little aloof as of late, be patient. You know how celebrities can be. First the commercials, then the six-page spreads in *major* publications, and now this. Yes, the Amiga is now the star of its own show. *The Amiga Video Magazine* began airing in November in the New York City area.

The half-hour program supports a magazine format and is devoted solely to the Amiga and its hardware, software, and their applications.

The Amiga Video Magazine concentrates on what the Amiga is achieving in the real world (aarrgh! not that!), traveling to places like production houses and recording studios where the Amiga is being employed in professional capacities. Each week, the show features different segments ranging from beginners pieces to advanced users segments. The program also devotes time to users groups.

Viewers in the New York City area can tune into *The Amiga Video Magazine* Mondays at 6:30 p.m. on Channel 23/J cable. The program will soon become available nationwide on the Arts and Entertainment network.

For additional information, contact:
Mark Montellese, Producer
(212) 724-0288

(continued)

At last, a scanner with an artist's touch.



With the Migraph Hand Scanner and Touch-Up™ you can now produce professional-quality high-resolution scanned graphics from start to finish—without changing programs.

This unbeatable hardware-software team lets you scan, edit, and enhance images until they're exactly right for all your desktop publishing projects.

Start with quality hardware.

The Hand Scanner has all the features you need: A scanning window over 4" wide. Four scanning resolutions—100, 200, true 300, and true 400 dots per inch. Adjustable contrast. Three dither settings for scanning photographs. Plus a special setting for line art.



You can build your own library of images from logos, photographs, books, and illustrations. And that's just for starters.

Finish with software that won't quit.

With Touch-Up, Migraph's complete design tool for high-resolution monochrome images, you can put the finishing touches on every image you scan. Its powerful editing functions include standard commands plus extras like rotate by degree, slant, stretch, outline, and bolding. A complete paint program. A full array of drawing tools. Not to mention special effects and scalable outline fonts. When your images are pixel-perfect, you can import them into your favorite Amiga publishing programs like Professional Page™ and Page Stream™. A variety of load/save formats also lets you use images on the PC, Mac, and ST.

Last but not least: The Migraph Hand Scanner and Touch-Up are easy to learn and easy to use.

See you dealer today for more details or call Migraph toll-free.

The Migraph Hand Scanner and Touch-Up. Powerful tools for professional publishing.

For Amiga 500, 1000 and 2000 systems with 1MB memory. A hard disk is recommended.

MIGRAPH INC.™

Migraph, Inc. 200 S. 333rd, Suite 220 Federal Way, WA 98003 (800) 223-3729 (10 to 5 PST) (206) 838-4677

© Copyright 1989 Migraph, Inc. The Migraph logo and Touch-Up are trademarks of Migraph, Inc. All other products named are trademarks of their respective companies.

Circle 138 on Reader Service card.

The Krueger Company can provide a wide variety of **MEMORY, MICROPROCESSOR, and SUPPORT CHIPS.**

ALL ***KRUEGER*** PARTS carry an unconditional 30 day guarantee. The minimum purchase is $50.00.

Call **Myron Lieberman** at **(800)245-2235** or FAX **(602)820-1707**. Mention *AMAZING COMPUTING.*

ANOTHER TIP FROM DR. CHIP



8 MHZ 68010	$ 1.00
12 MHZ 68020	**$30.00**
16 MHZ 68030	**$40.00**
5 MHZ 8088	$ 1.00
8 MHZ 8086	$ 1.50
8 MHZ 80286	**$10.00**
DP8464BV3 PLCC	$ 1.00

ASK ABOUT **Z80**s AT GREAT PRICES

The Krueger Company **(800)245-2235.**

Circle 118 on Reader Service card.

STEEL PAWS

Well, if you've noticed any long lines streaming out of your local Ticketron dealer, it's probably folks lining up to witness the Performance of the year. No, it's not some monumental rock band kicking into gear for a North American tour. This Performance comes from Pregnant Badger Music.

Performance is a new database program that allows MIDI musicians to incorporate existing patch files and sequences into live situations. Performance is designed to allow the performing musician to keep track of MIDI information including Synth Presets, Patch Banks, and Sequences. Performance also works by sending MIDI information to your instruments, thus automating your MIDI setup.

Performance features a built-in set editor, as well as a built-in sequence player that plays Standard MIDI Files. The program includes a user-configurable database, the complexity of which is limited only by the amount of available memory. Performance supports interlace and non-interlace screen modes, and a large font legible from up to 30 feet away.

The Pregnant Badger program runs on any Amiga and requires 1 M memory, AmigaDOS 1.2 or higher, and a MIDI Interface. The Performance package includes one program disk and a 95-page manual.

Performance
Pregnant Badger Music
10010 Biscanewoods Way
Sacramento, CA 95827
(916) 361-8217
Price: $165.00
Inquiry #231

THE NEW AND IMPROVED

From West Chester comes news of new products, prices, and PR. Commodore has introduced the Amiga 2500/30 into the Amiga computer family. The latest Amiga features a 68030 processor and a 68882 math coprocessor, as well as two bridgeboards that allow A2500/30 users to run both MS-DOS hardware and software. According to Commodore Director of Business Markets David Archambault, Commodore developed the A2500/30 to "meet the demands of [their] customers and the increasingly sophisticated applications being produced by Amiga developers."

Suggested retail price for the A2500/30 is $4,699. Owners of the 2000-line Amigas can upgrade by purchasing the A2630, a newly available 68030-based accelerator board. The suggested retail price for the A2630 is $2,195.

In addition to the A2630, Commodore announced the availability of several other new Amiga peripherals, including the A2091, an autobooting hard disk controller and memory expansion board, and the 2091-40, an autobooting 40MB SCSI hard disk and controller. Also new is the A2232, a multi-serial port card which provides users with seven additional RS232 ports, each capable of operating independently or simultaneously, providing speeds of up to 19.2 KBaud. The A2091 and A2091-40 retail for $399 and $899, respectively. The A2232 carries a retail price tag of $399.

But before you start rolling your pennies for that A2500/30 or A2630, you should be aware that Commodore also announced price breaks on some of their 2000-line products. The A2000, which formerly retailed for $2,195, will now retail for $1,899. The retail price of the A2000HD has dropped from $2,999 to $2,699. The A2500/20 is down to $3,999 from $4,699 retail. And the A2620 card, which formerly retailed for $1,995, will now go for $1,494.

In other Commodore news, Commodore has formed a multimedia marketing group to be headed by former Disney producer Ken Christie. With ten years of marketing and creative experience in the video production and interactive video industries, this fellow is — pardon the expression — no Mickey Mouse. As part of an effort to enhance Commodore's position in the multimedia industry, Christie will reportedly be responsible for the management and integration of Amiga marketing programs in desktop and professional video, interactive video, and professional music.

Commodore Business Machines, Inc.
1200 Wilson Drive
West Chester, PA 19380
(215) 431-9100
Inquiry #232

•AC•

Snapshot

- Star Command
- Super Hang-On
- RoboCop
- Bad Dudes

by R. Bradley Andrews

STAR COMMAND

First on the list this month is Star Command, a game recently ported to the Amiga for Strategic Simulations by Andromeda Software. The game takes place in the far future in a remote corner of the galaxy. The earth was destroyed long ago by a hostile alien race, but humanity is still a force to be reckoned with. Humans are concentrated in a section of space known as the triangle, an area of space inside the protective border of the three main defensive starports. The immediate area around the triangle is divided into roughly two camps. The Alpha Frontier was an extensive mining area but has become infested with pirates. The Beta Frontier is occupied by a race of hostile intelligent insects. Beyond these are many other races, but only a robotic race is known to have the same level of technology. Fortunately, they never venture into the triangle itself.

You must handpick the eight team members you will direct on missions throughout the galaxy. Characters are rolled randomly and can specialize in one of four different fields: Marine, Soldier, Pilot, or Esper. The first two are variances of grunt soldiers and will do most of the combat. Pilots fly the spacecraft and jumpships that take your squad from planet to planet. The final category, Esper, is only open to those who are highly skilled in mental force projection.

Each raw recruit must first complete eight years of basic training before becoming eligible for squad membership. You control the assignment choices for each of these years. Assignments will increase skill levels in the appropriate area, increase a character's own statistics (e.g., make him quicker), or advance him on to the next rank. Ranks are important, since they determine the amount of pay a character receives for performing duties. While there is a random chance of gaining a rank during any assignment, only Officer School guarantees one rank of advancement (though it does nothing else). The chosen category is not always available and sometimes a character is assigned to a category other than the one chosen. Generally it is not too bad, but if the alternate assignment is a desk job, that year has just been wasted. The game encourages you to spread this training out a bit since no skill level can be increased more than three times.



Super Hang-On puts you in control of one of the fastest cycles in the world.



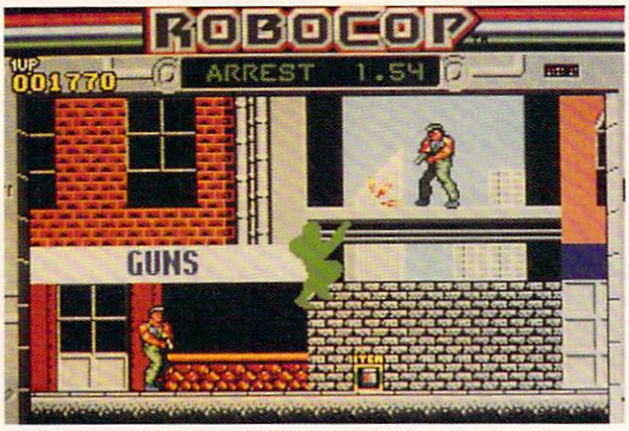
Star Command: role playing in a remote corner of the galaxy.

After the squad is created, it must carry out the missions assigned to it by headquarters. While missions do vary somewhat from game to game, a general pattern is followed and each must be completed before the next is assigned. Missions begin simple but grow increasingly difficult until the final climactic mission which — you guessed it — involves saving humanity.

Combat is the main focus of the game. Both space and ground battles will occupy most of your time. Each involves a partially simultaneous move-and-fire sequence. Depending on the speed of the men and the ships fighting, each side will get up to six phases to move, which will be followed by a combat segment. Immediately prior to combat, you must select the opponents for each of your squad members. Then the computer will play out each of the actions in order, once again based on the speed of the combatants. It is possible to disengage from most combats by either moving out of range of all opponents in the case of ship-to-ship combat, or leaving via an exit during a man-to-man encounter. Some actions, such as boarding enemy ships, do not allow for leaving, so it is a fight to the death.

(continued)

If the game sounds interesting so far, don't get your hopes up too high. SSI has yet to learn how to do a truly good Amiga port. The graphics, sound, and user interface are far below par. The graphics look almost like a straight copy from IBM EGA graphics and come nowhere near the richness possible on the Amiga.



RoboCop: serving and protecting the citizens of the future.

The interface is very character-oriented; I never had the need, or ability, to use the mouse in the program. Though it remains visible the entire time, I often had to "hide" the cursor in the bottom left corner so it would not obstruct my view of the screen. The game incorporates extensive use of menus, but not the kind Amigans are familar with. They are common on IBM programs for users without a mouse or other pointing device. Typing the appropriate letter for an option either performs that action or takes you to a submenu with further options.

I guess this would not have been so bad if the menus had been consistent, but they are not. In one place, selecting "G" may change the time delay for message display; in another, selecting "T" may do this; still in a third instance, "C" may be the proper key to press. The system was not well thought out.

My other main gripe about the interface is its slowness when changing from mode to mode, such as initiating combat or entering a starport, and the difficulty of getting even simple information, such as detailed squad statistics. It can be very difficult to be sure everyone is armed properly, or to find exactly who has the flamethrower.

The graphics have a bland look about them. While appropriate for much of the information, boring text is used throughout, and it is apparent that graphical richness was not even considered during development. Sound is adequate and on the same level as the graphics, and that isn't saying much.

In case you haven't realized, I am extremely annoyed with many aspects of the game. Why not just dump it and write about something else? Believe me, I almost did. I became extremely frustrated at the start and almost quit. But I persevered and after many hours of play, I witnessed gleanings of a reasonable game system underneath.

It is fun to build up a squad from lowly Privates to Grand Admirals during the game (though I never got that far). The missions, while often somewhat silly, do tie the game together. There just isn't that much out there for those who like "role-playing" games and don't want to play another D&D. So I kept coming back to the game until I finally finished the first main mission, finding and defeating the pirate Blackbeard. I was tempted to waste even more of my weekend, until I sat back and asked whether the game was really worth it. I decided I had played enough to write about the game and put it away permanently. I just wish someone would come out with a good game with a similar theme. It would be wonderful if SSI would rewrite the game to make it more playable and enjoyable, but I will not hold my breath waiting. Perhaps I will have the opportunity to work on a space game of my own shortly....

In closing, let me say that I can only recommend Star Command for those who are literally starved for a space role-playing game. If you have the ability to ignore the poor graphics and the patience to work through the painful interface, you may actually enjoy the game. But otherwise, I would steer clear.

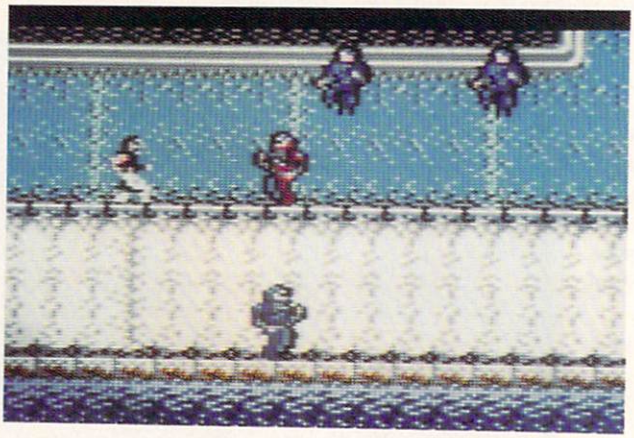
Next this month are three new arcade game translations from Data East. Each is an adaptation of an arcade favorite.

SUPER HANG-ON

Super Hang-On is a motorcycle racing game that puts you in control of one of the fastest cycles in the world. Only your skill and reflexes will allow you to win the ultimate race. The race is split into four racing segments of differing difficulty levels. Africa is the easiest, with only six checkpoints required for the entire race. The European race is the most difficult and has eighteen checkpoints to pass through. As with most games of this genre, you begin with a fixed amount of time at the start and gain additional time bonuses for successfully crossing each checkpoint. If time runs out before you cross the finish line, the game is over.

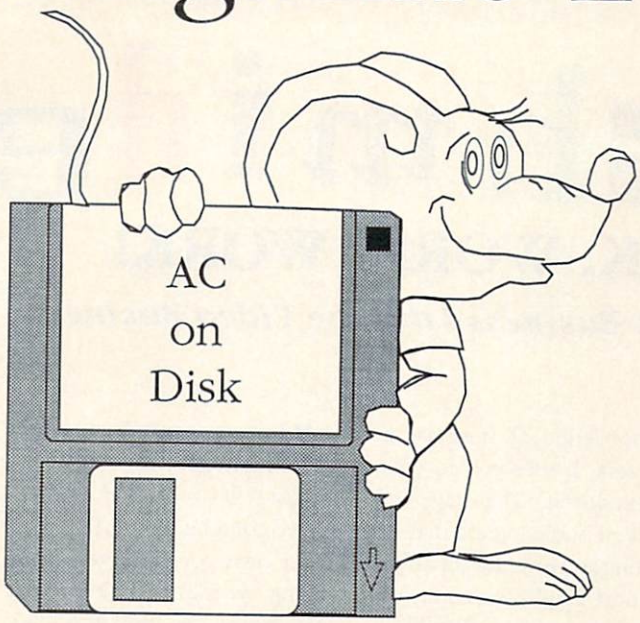
Many other racers crowd the track and must be carefully avoided. Even slightly bumping them will drastically slow

(continued on page 22)



Bad Dudes: Fighting your way through Ninja-infested sewers in search of high-ranking officials.

Enjoy Your Favorite Magazine Even More.



Order AC's Disks and save yourself hours of typing.

AC's Disks include source code, executable, and IFF pictures, when available, for all articles that are printed in *Amazing Computing*.

AC Disk #7

This disk contains the source and executable code relating to articles in AC V4.11 & AC V4.12

Arexx Part II: Inofrmation on how to set up your own ARexx programs with examples. Author:Steve Gilmor.

Leggo My LOGO: A Logo program that generates a Christmas tree with decorations. Author: Mike Morrison.

Trees and Recursion: An introduction to binary trees and how to use recursion. Written in C. Author: Forest Arnold.

C Notes V4.12: A look at two data compressing techniques in C. Author: Stephen Kemp.

Animation? BASICally: Using cell animation with AmigaBASIC. Author: Mike Morrison

Menu Builder: A utility to help build menus in your own programs. Written in C. Author: Tony Preston.

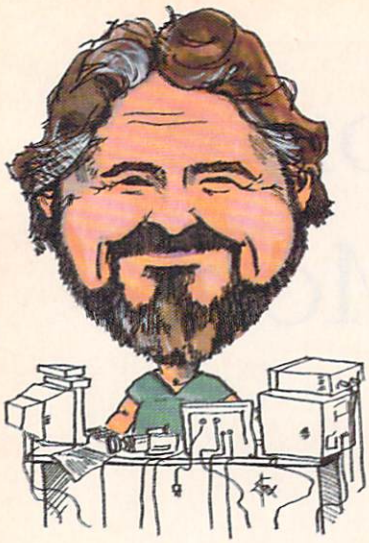
Dual Demo: How to use dual playfields to make your own arcade games. Written in C. Author: Thomas Eshelman.

Scanning the Screen: Part four in the fractals series. This article covers drawing to the screen. In AmigaBASIC and TrueBasic. Author: Paul Castonguay.

C Notes V5.1: Recursive functions in C. Author: Stephen Kemp.

For a complete listing of AC disks, see page 106.

Only $6.00 per disk ($7.00 for Non-Scubscribers).



video Schmideo

WORK, WORK, WORK!

or: There's No Business Like The Video Business

by Barry Solomon
Video Editor

I know a lot of you Amigans out there have a secret yearning. Some of you have had your Amigas a few days; some a few weeks; some a few months or years. Some of you haven't even bought your Amigas yet. But you all have one thing in common: you want to be in the video business!

If this is your secret yearning, I have some tips for you. Whether you are a complete novice or an accomplished Amiga artist or animator, if you want this to be more than a hobby, if you really want to make it big in the video biz: **THINK OF IT AS A BUSINESS!**

This may sound a little too simplistic, but it isn't as dumb as it sounds. Every year thousands of hobbyists try to turn their hobbies into businesses. The majority of them fail miserably. The statistics on new business failures are scary. Usually these entrepreneurs fail not so much because they are bad businesspeople, but because they just can't think of their hobby as a business.

Businesses usually require a great deal of thought, planning, and patience. Not to mention money, a little daring, and some luck. The Desktop Video business is no exception. You may know everything there is to to know about producing incredible videos, and you may have equipment that would make any TV station salivate, but can you run a business? Have you ever been in business? What do you know about running your own business?

If your knowledge of business is rusty or nonexistent, I refer you to your local library or bookstore. There are any number of excellent books on the market dealing with starting and running your own business. Also most local community or junior colleges have courses on this topic.

Every year thousands of hobbyists try to turn their hobbies into businesses. The majority of them fail miserably. The statistics on new business failures are scary.

To begin with, you need a business plan. A good business plan will not only contain lots of numbers (expenses, investments, projected income, etc.), but also answers to the following questions:

1. What services will you offer?

Exactly what is it you are proposing to do? Will you be doing graphics for other video companies, or will you be producing your own complete videos? Will you be doing weddings or industrial videos or commercials for broadcast TV?

2. To whom will you offer your services?

Who exactly will be your clients? Friends? Neighbors? Corporations? School systems? Ad agencies?

3. Is there a real market for these services in your area?

Once you've decided what you want to do and for whom you want to do it, you must ask yourself if there is anyone like this in your area who really wants and needs this service. And will they be willing to pay for it?

4. How will you sell/advertise/market these services?

How will you make these people aware that you are in this business? How will you make them choose you over the competition?

5. Do you have any competition?

Are there already people (competitors) in your area offering these services? If there are it doesn't mean that you can't also be successful, but you now have to exam-

ine a number of other questions. Who are these competitors? What exactly do they do? How well-known are they? How good are they? What do they charge? Can you compete in terms of quality? In terms of price? Is there enough potential business for all of you?

Anyway, you probably get the picture. You can find lots of information on developing a business plan at your local library. Better yet, check with your local Chamber of Commerce, Businessperson's Association, SCORE (Service Core of Retired Executives), or even the Small Business Administration office. They can often give you the information you need along with key advice.

Don't be surprised, however, if you find that none of these people have ever heard of Desktop Video! It will probably be up to you to explain this to them and to dig out the generic business information from what they give you. Also, do not be too surprised if they do everything they can to talk you out of it: if this is a business they have never heard of, they will almost certainly be afraid of it.

Don't forget, also, that your area will greatly affect your business plans. If you live in New York City or Los Angeles you already have lots of competition! Don't expect to make it big there unless you have oodles of money, considerable talent, or some other factor which will make you stand out in the crowd.You may have to market your services outside of your immediate area in order to compete at all. If you live in a very small town, you may find that there is just no market in your area for what you'd like to do. Here again you may have to pitch your services outside your region to survive.

If you're already in the video business and just want to begin adding computer graphics, titling or animation, life will be a little easier. If you're starting from scratch and you do not know computers or video, life will be a lot tougher.

For those of you who are doing neat things with your Amigas but have not yet tried to get into video, I have a BIG tip for you (no thanks necessary, just send cash).

Before you start buying cameras, VCR's etc., and try to learn how to use them, think about this: How about providing those incredible computer graphics you're doing for someone who's already established and successful in the video business?!

Look up video services in your local Yellow Pages. Check with your local cable station. Call your Uncle Fred's neighbor who videotapes weddings. If you can hook up with one or more people already in the video biz, you may find yourself a steady stream of work!

The advantages to this approach are immense. You may need to do little or no marketing of your services beyond your initial contacts. You may also have saved yourself hundreds or thousands of dollars investment in equipment! Most of these people will either have or have access to editing equipment which can be used to combine your graphics with their video. Often they will come to you with portable equipment to record your output. Once they have your Amiga output on tape they can use it as they wish. You are finished, paid (hopefully), and on to the next project! All you really need for this, besides your Amiga and software, is a genlock or encoder! This way you have spent minimal time with a minimal investment. The rewards can be very nice.

Whether you are a complete novice or an accomplished Amiga artist or animator; if you want this to be more than a hobby; if you really want to make it big in the video biz: ***THINK OF IT AS A BUSINESS!***

If you really want to do the video part yourself and you are starting from scratch, you have a lot to learn. Answering those questions in your business plan completely and honestly will help you determine exactly what equipment you'll need. Research that equipment thoroughly before you spend a penny. Take your time. Nothing is more frustrating (or more expensive) than buying the wrong equipment. Especially when you discover the fact three or four months down the road in the middle of a project!

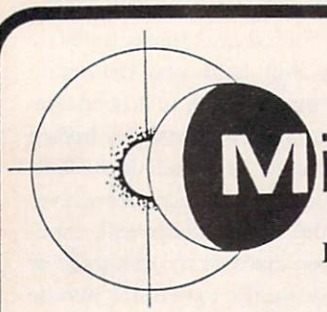
Be prepared to put some serious time and effort into learning about video. Many local community colleges now offer video classes. Or check with your local cable company. Many of them offer basic video instruction as part of their public access programs. Just having taped your friends and/or relatives at a party does not qualify you as a videographer (yes, that's the term).

Watch a lot of TV. (Finally, the excuse you've been waiting for!) Make notes about or tape the professional graphics that you admire. Study them to see what the pros are doing. Get ahold of the videotape or videodisc called "The Dream Machine" to see the current state of the art in computer graphics. If you haven't seen this before, and you're really into computer graphics, be prepared to do some serious drooling. Even if fancy 3D ray-traced graphics are not what you're planning on doing, it's still important to know what can be done. You should never be ignorant of what the current state-of-the-art is in your own field, even if your own work will be at a much less complicated level.

Finally, you must understand going in that if you're very proficient at what you do, have a lot of luck and work very hard, you may be able to do OK. However, you'll never make it big in the video biz without a lot of talent and imagination!

This month in my Video Schmideo mailbag I received a letter from Kenya Pittman of Philadelphia. Kenya wrote regarding my first article (AC V4.9). Apparently, my obsession with desktop video sounded familiar and Kenya, having a similar drive, wants to know what equipment to purchase. Specific questions included: "Should I buy a 500 or a 2000?", "What type of genlock?", "How much memory?", etc., etc.

Well Kenya, I'll do what I can. First of all, even without knowing exactly what



MicroMiga

Discount Hardware & Software

Orders Only 1-800-733-AMIGA

P.O. Box 2104, La Mesa, CA 92044
Customer Service 1-619-670-3161
24 Hour FAX 1-619-670-9732
BBS 1-619-670-1095

NO CHARGE FOR VISA & MASTERCARD

Hardware

ASDG ScanLab	$815
Amax Mac Emulator	$128
Boing Mouse	$99
California Access 880	$133
Digi-View Gold	$135
Dual Serial Port Board	$211
Expansion Tech Toolbox (500/1000)	$215
GVP A3001 w/68030/68882/4mb RAM	$2,500
Kronos Hard Disk Controller	$211
Magni 4004 Genlock w/Controller	$1,685
MicroBotics Hardframe DMA	$250
Panasonic WV-1410 Camera w/Lens	$234
Scanlock Genlock	$872
SupraDrive 500 30meg Hard Disk	$614
SupraRam 500 (512k. Clock/Cal)	$99
SupraModem 2400 External	$119
SupraModem 2400zi Internal	$126

Software

DataStorm	$26
Deluxe Paint III	$105
Digi-Paint 3	$65
F-16 Falcon	$32
HiSoft Basic	$107
Lattice C 5.02	$190
Lords..Rising Sun	$33
Music X	$175
PageStream	$129
Pen Pal	$88
Populous	$35
Professional Page 1.3	$175
Sculpt/Animate 4D	$386
Superback	$51
Superbase Professional	$225
The Works! Platinum	$175
World Atlas	$39

- Toll Free Ordering
- Free Catalog
- Same Day Shipping
- FAX Ordering
- Credit Cards Not Charged Until We Ship
- We Carry Over 1300 Titles

Circle 182 on Reader Service card.

At MicroMiga our customer and their Amiga are Number One. We carry over 1300 Amiga products, WAY too many to list here, discounted up to 35% off retail! Defective items repaired or replaced within 10 days of delivery. Non defects subject to 20% restocking fee. Shipping charges are $4.50 first item and $1.00 for each additional item. Hardware shipping rates by quote only.

you would like to do, I would have a very tough time recommending an Amiga 500 for professional use. Don't get me wrong, it's the same great machine that the 2000 is, basically, and it is expandable, but a 2000 (series) will probably save you a lot of headaches down the road. Besides which, the 2XXX machines also have the video slot which is necessary for some genlocks and other forthcoming video devices (the Video Toaster and others come to mind).

At this point, it may also be important to think of speed. If you would like to do 3D ray-traced animations and you would like to do many of them before you get old, you might want to get a faster machine. You can add faster processors from Commodore or third-party suppliers to your 2000 or, if you don't feel that you need the speed as yet, just start with a 2000 to make the upgrade option easier at a later date.

The 500 can be similarly upgraded, but it is more difficult and, in some cases, more expensive. Alternately, you can go straight to the top and start with the Amiga 2500 (with a built-in 68020 processor and 68881 math coprocessor), or go to the brand new 2500/30 which includes a 68030 board running at 25 MHz as well as a 25 MHz 68882 coprocessor.

As far as memory goes, I honestly don't think there is such a thing as "enough". I have a 2000 with a total of 3 meg (at this point) and, frankly, I run out of memory fairly often. The amount of memory that you absolutely must have will depend very specifically on exactly what you intend to do.

My best blanket recommendation, especially if you want to do anything in animation, would be 3 meg. Also please do make sure that, whichever machine you purchase, you do get the new 1 meg Fat Agnus chip. All current models have been shipping this way for a couple of months, but there are probably a few without them left on the dealers' shelves.

As far as a genlock goes, please check out my column in AC's November issue (AC V4.11) so I won't have to quote myself at length and appear pompous. Just remember (I'll paraphrase) that no matter how good your work is, it will be worthless unless you can get it out of your Amiga and onto video with the quality you need (and your clients expect).

As far as other hardware "necessities" are concerned, again, you must decide what you want to do. I have found a digitizer to be indispensable. I am also currently using a scanner (to digitize logos and such), which I am beginning to feel I cannot live without.

Once again, that's about it from this end. Please don't hesitate to write me if you have any video problems or questions that require common-sense answers.

Barry Solomon
Video Editor
c/o Amazing Computing
P.O. Box 869
Fall River, MA 02722

•AC•



Amiga Users Groups



F.A.C.E.
Fayetteville Amiga Community Exchange
1014 Streamway Drive
Spring Lake, NC 28390
John Berger, Secretary

Meeting: 3rd Wednesday of each month at 7 p.m.

Location: Software Exchange, Sycamore Square Shopping Center, Fayetteville, NC.

BBS: We use the BRAGG BBS; Phone: (419) 432-2013

■■■■■■■■■■

H.A.U.G.
Hartford Amiga Users Group
P.O. Box 504
South Glastonbury, CT 06073
Bob Morrison, Treasurer

Description: Organized 12/86, with 70+ active members, monthly meetings with demos and presentations, a club newsletter, free monthly club disk, complete public domain library, and SIG's.

Meeting: 3rd Thursday of each month at 7p.m.

Location: Aetna Institute, Farmington Avenue at Sigourney Street, Hartford, CT.

BBS: Under development.

■■■■■■■■■■

P.A.U.G.
Philadelphia Amiga Users Group
P.O. Box 21186
Philadelphia, PA 19154
Tony Antonuccio, Editor

Description: The largest Amiga users group in the metropolitan Philadelphia area. We are eager to help new Amiga users get the information they need.

Meeting: 2nd Saturday of the month at 12:30 p.m.

Location: Drexel University, MacCallister Hall, Room 4014.

BBS: We have a working relationship with PhilAmiga BBS; Phone: (215) 533-3191.

Newsletter: The Blitter, monthly.

■■■■■■■■■■

R.G.V.A.U.G.
Rio Grande Valley Amiga Users Group
c/o Marilyn Martin
Computer Graphics Lab, NMSU
P.O. Box 30001, Dept 3AT
Las Cruces, NM 88003
(505) 646-1840

Description: Our group is small but intensely involved with animation.

Meeting: 1st Thursday of each month at 7 p.m.

Location: Room 129 Jacobs Hall, NMSU, Las Cruces, NM.

■■■■■■■■■■

BYU Amiga Users Group
432 ELWC
Brigham Young University
Provo, Utah 84604
Brian Koetting, President
(801) 377-2269

Description: Group members are principally students of Brigham Young University, but meetings are open to all. PD sharing, projects, and workshops on the many applications of the Amiga is the group's focus.

Meeting: Every 2nd and 4th Saturday at 10:00 a.m.

Location: 258 ELWC on the Brigham Young University campus.

■■■■■■■■■■

C.D.A.U.G.
Capital District Amiga Users Group
P.O. Box 14353
Albany, NY 12212
Carmen Artino, Secretary
(518) 783-1784

Description: We are loacted in Albany, NY but serve Amiga computer users in the entire Capital District of New York State. We are a very active group and our meetings are very well attended. We have active subgroups with varied special interest (video, graphics, music, programming, business, etc.).

Meeting: 3rd Thursday of the month at 6:30 p.m.

Location: The Computer Cellar, Westgate Shopping Plaza, Albany, NY.

BBS: Atredes BBS, 1 line, 80-meg HD, Amiga 500, 1 meg RAM. Phone: (518) 370-5207. Sysop: Lou Matrazzo

■■■■■■■■■■

V.C.C.
Valley Computer Club
P.O. Box 310
Denair, CA 95316
Ruby Larson, Secretary
(209) 529-8473

Description: Supporting Amiga computers in Turlock, Modesto, and the surrounding California area.

Meetings: 3rd Wednesday of the month at 7 p.m. to 9 p.m.

Locations: Doctors Medical Center Conference Room, Florida and Orangeburg Avenue, Modesto, CA.

Newsletter: Valley Computer Club

■■■■■■■■■■

T.C.U.G.
Textile Computer Users Group
P.O. Box 1065
Lafayette, CA 94549
Margaret Copeland, President
(415) 284-1635

Newsletter Announcement: The Textile Computer Users Group announces the publication of Computer Textile Exchange. CTE is dedicated to providing the professional and amateur textile designer with the latest information for using personal computers in textile design. Quarterly, $24/yr.

•AC•

(Snapshot, continued from page 16)
your speed down, hindering your progress. For some reason, they are not affected at all by a crash with you; this can be somewhat annoying.

Obstacles also appear at frequent intervals along the sides of the road and hitting these can not only throw you for a loop, but also cost valuable time while you remount and restart your cycle.

Super Hang-On features a subjective view of the area in front of your cycle. Objects zoom past based on your current speed. One glaring deficiency is the lack of any kind of rear-view mirror. While you are usually focusing on passing other cycles, opposing cycles will pass you if you are going slow enough. This may be realistic, but without a way to tell where they are passing you from, it is very easy to be hit by several cycles in a row when regaining speed lost in a collision, slowing you to a virtual halt after each collision.

Either a mouse, a joystick, or the keyboard can be used to control the cycle, but I found the joystick to be the preferable method. The game has a few options that can be adjusted for play, such as the sensitivity of the control device and the music to play during the trip. Each race has its own backdrop based on the prevalent terrain type.

ROBOCOP

RoboCop is an arcade game based on the policeman of the future. RoboCop has three missions: 1) Serve the public trust, 2) Uphold the law, and 3) Protect the innocent. To do this, he is authorized to use whatever force necessary to free the streets of Old Detroit and clean up the Scum who have tried to take over. Armed with a variety of weapons, including a Special Issue Auto-9, you must accomplish all this without letting RoboCop's power level be drained too low.

Nine levels in all await to challenge you, with a few intermission "bonus" rounds thrown in for fun. Each will require a slightly different strategy to complete while keeping yourself intact. The play segments use a standard horizontal scrolling routine and most of your movement is from left to right; however, some levels modify this a bit and may even require you to go down a few levels to finish the segment. The joystick is the primary control device and is used to move RoboCop around the screen, with the fire button firing the gun, or punching in the current direction.

BAD DUDES

The third game is Bad Dudes. It seems Ninja criminals are everywhere. Even the White House is not safe. The President has been kidnapped by the evil Dragon Ninja, and only you can help.

Beginning in the city, you must fight your way through the sewer, forest, freight train, and cave on your way to the factory where the President is being held. At the end of each segment, you must defeat that level's super bad guy to advance to the next challenge.

Should you somehow manage to make it through all seven levels and defeat the Dragon Ninja at the end, your quest will be completed, and you will be a hero to the free world.

Once again, the joystick is the primary control device, but this time it does a few more things. Not only is the joystick used for simple left and right movement, but pushing it in the other directions with and without the button will cause one of the many possible karate attacks to be carried out. This can be a bit difficult to learn and will require some time to master.

The graphics in all three games are reasonably close to the arcade original, with Bad Dudes being the best, RoboCop coming in second, and Super Hang-On a distant third. Racing games have been done better, but Bad Dudes has some pretty sharp graphics for all the action that can occur on the screen at one time.

The sounds are also faithful to the originals. Again, the ranking is the same, with Bad Dudes' effects being the most realistic, but any limitation with the other two games is more a limitation of the arcade original. The background music can become annoying in Super Hang-On, but since the music can be turned off, this is not too much of a problem. The first two both feature digitized start-up sounds, one of a motorcycle revving up and the other of the recitation, in true movie style, of RoboCop's missions.

All three lack a demo play mode. This is disappointing as it is usually fun to watch the computer play through at least the first few levels to get an idea of exactly what is going on.

Also, none of these games save the high score to disk. Such a waste. Who cares what the high score was during my most recent fifteen-minute playing session? While the price seems a bit steep, these games might be worthwhile if it were not for some new competition.

HIGHS AND LOWS

The most recent release from Innerprise software, Battle Squadron, is bound to set new standards for Amiga games. Programmed by the same programmers who brought us Hybris, the sound and fast-paced action are very nice. The game does have a few flaws, such as the inability to continue in some parts of the game, and the usual failure to save the high score to disk. But this is a game every serious arcade player should have.

Finally this month, I want to steer you away from buying Dr. Doom's Revenge, by Paragon Software. While the premise is interesting (controlling both Spiderman and Captain America in their quest to remove a stolen nuclear weapon from Dr. Doom's evil control), the game mechanics are downright lousy. It takes an extremely long time to even start up the game, and then the controls are so difficult to master that you will quickly die when facing your first robotic foe, let alone any real super villain.

Most bad games at least let you live through the first few minutes of gameplay. Avoid Dr. Doom's Revenge at all costs and spend your money on something else. See you all next month.

•AC•

Star Command
Strategic Simulations
550 South Winchester Boulevard,
Suite 200
San Jose, CA 95128
Price: $19.95
Inquiry #209

Super Hang-On
C/O Data East USA, Inc.
1850 Little Orchard Street
San Jose, CA 95125
(408)286-7074
Price: $44.95
Inquiry #210

RoboCop
C/O Data East USA, Inc.
1850 Little Orchard Street
San Jose, CA 95125
(408)286-7074
Price: $44.95
Inquiry #211

Bad Dudes
C/O Data East USA, Inc.
1850 Little Orchard Street
San Jose, CA 95125
(408)286-7074
Price: $44.95
Inquiry #212

MENU BUILDER!

by Tony Preston

This article is about menus. It is also about making the job of creating them much easier. I am not only going to describe how to build them, but I will also describe how to automate the process!

The Intuition manual describes menus by telling you how to set up the needed structures. I may be a little slow, but I gave up on menus after rereading that chapter about three times! It was not that things were hard to understand, but the documentation just doesn't have many examples. It would have been nice if the window example shown in the ROM Kernel manual had been repeated with an example menu! To understand and use menus, you have to be familiar with several other things in Intuition. When I first looked at menus, I was just learning the Amiga and I didn't realize that menus are not as hard as they appear!

When I finally got back to menus about a year later, I found that things were not as complicated as they initially seemed! It seemed that Menu, MenuItem, and IntuiText structures could be setup with default values making things simpler.

Since I am a lazy person, (a fact my wife will verify!) I decided to write a program that would automate the job. Looking at the examples and instructions in Chapter 6 of the Intuition Manual, I realized that much of the menu building process could be calculated very simply. For example, menus could be placed on a window based on the width of the title. The width is in pixels which is easily calculated. The width of the menu items could also be calculated using the longest item as the base width for the whole menu. Many other options and parameters could be preset to constants in most cases. I wanted a simple text menu (no graphics) with support for key stroke equivalents. After I thought about the problem, I came up with the MenuBuilder. A program that takes a simple text description and creates all the needed structures. As you know, I'm a lazy person, and I like to set up the menu with as little work as possible. Then take output from the menubuilder and use it with as little work as possible. The program has to generate all the needed structures and make the resulting source file independent of other files, so I can compile it by itself. That way, if I want to change the menus, I only have to rerun the menubuilder, compile, and relink. Before we get to the menubuilder, let me describe how a menu is structured. The example here will be similar to the Workbench menu. Menus are always associated with a window. A program always attaches a menu to a window with a call to the system routine "SetMenuStrip". If a program has multiple menus to use, it must first clear the old menu before adding the new one. A menu is cleared by using a call to the system routine "ClearMenuStrip". This all seems rather simple, right? The basic sequence you should always use is:

```
- OpenWindow

-...paired sets of

- SetMenuStrip

- ClearMenuStrip

- CloseWindow
```

It is important to remember that you have to remove the menu BEFORE you close the window. The system usually calls up the Guru to tell you that you forgot to remove the menu!

SetMenuStrip takes a window pointer and a menu structure pointer as arguments. Here is a simple example of code that opens a window and attaches a menu to it:

```
extern struct Menu *MyMenu;

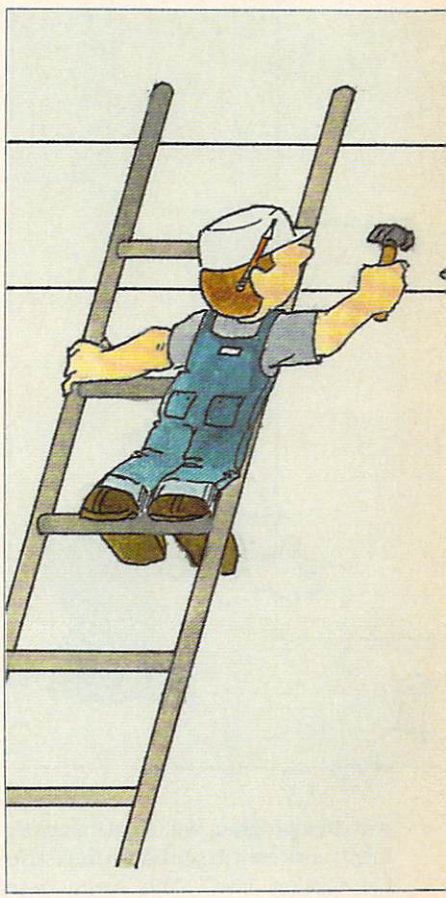
struct Window *Window;          /* ptr to window*/

Window = (struct Window *)OpenWindow(&NewWindow);

SetMenuStrip(Window, MyMenu);

ClearMenuStrip(Window);         /* Remove Menu */

CloseWindow(Window);            /* close window */
```



The structure pointed out by the external pointer MyMenu is the one built by the MenuBuilder! This means that code similar to the above just has to be linked again to use updated menus, at least, if the changes are only positional.

The way you act on a menu event is to wait on the window's port, get the message sent to it and discover that it is a MENUPICK item. Messages are an interesting thing on the Amiga. You can get all kinds of different messages and each one requires that you reply. Waiting on a Port is the proper way for an Amiga program to wait for user input. You should never code a loop that checks a device directly, using a CPU delay. A device should generate an interrupt that makes your task ready by sending it a message to

your port! This is what happens when you click the mouse buttons! The code for a simple loop that waits for a MENUPICK item and properly replies to all messages sent to your task looks something like:

```
for (;;)   /* loop until a close gadget  */
{

    WaitPort(Window->UserPort);

    message = GetMsg(Window->UserPort);

    if((message->Class) == MENUPICK )
        {

        M0 = MENUNUM(message->Code);

        I0 = ITEMNUM(message->Code);

        S0 = SUBNUM(message->Code);
        }

    else break;

ReplyMsg((struct Message *)message);
};

ReplyMsg((struct Message *)message);
```

The MENUNUM, ITEMNUM, and SUBNUM code above are macros that do the magic shifting and masking to the Code item to decode the menu, item, and sub-item number from the message sent by Intuition (in Intuition.h). The above assumes your window only has CLOSEWINDOW and MENUPICK flags set. Because of that assumption, Intuition will only send you the Menu selection and Close Gadget message. That is why the code above ASSUMES that if it isn't a menu message, it must be the Close Gadget! It becomes a simple coding exercise to act on each menu item as it is selected. This is all fine, except that the menu structure for even a simple menu must have three different structures, positioning information for each item, and the external pointer, MyMenu, must be defined for this all to work. Listing One is the full source code for this simple program. It will open a screen and a window, attach the menu, and receive the menu events until you click on the close gadget. The MenuBuilder along with this program will create a test driver where you can create a menu and test it to see how it looks! Now lets look at a sample menu file (Listing Two). This menu description contains the menu example that imitates the Workbench. The Menu also includes an extra menu thrown in to show sub-items. MenuBuilder decodes the commands MENU, ITEM, and SUBI to create the needed data to produce the menu structures you need to compile (Listing Three). There are many options to menus and IntuiText structures. Most of them we will not need in this article. The menu example here is a simple text menu. With menus, you can do much more than just text menus, but I will leave those areas as an exercise to the reader (I always wanted to say that!).



The menu description consists of the three command types. The syntax of the commands are:

```
MENU,Name

ITEM,Name,[Select],[Command]

SUBI,Name,[Select],[Command]
```

Where:

Name is the name of the Menu, item, or Sub item.

Select is an optional item which will display when the item is in position to be selected. Intuition will overlay the text in Select field on top of the text in the Name field.

Command is the Character to use for selection of the menu item with the Right-Amiga key. No check will be made to see that characters are not used more than once.

The MENU command defines a menu. The name will be the heading on the bar. The MENU command is the first command describing your menu. It becomes the heading! Simple right! That's the whole idea. You might be wondering where the heading will be positioned. The MenuBuilder will take care of where it will be positioned. You just tell it what headings you want!

After each MENU comes the actual items in the menu. To specify the items you use an ITEM command. Say, this is getting pretty easy isn't it! The ITEM command has a Name, an optional selected name, and an optional Command character. The Name will display on the menu when it is dropped. The Selected name will replace the Name when the mouse passes over the item. The Command character will be the character you use with the Right-Amiga key to select the menu item.

Each Item may have sub-menus. This is one of the trickiest parts of creating menus. You have to overlap the Sub-Menu with the item it belongs too. If they do not overlap, you cannot select the sub-item with the mouse. The MenuBuilder overlaps the Item by about 25% (the bottom right hand corner). It will calculate all the positions and make sure Intuition is happy!

To use the MenuBuilder, you just create a file like the one shown in Listing Two. I called the file "Listing2.menu" for this example. Each menu, item, and sub-item is described and then you invoke the menubuilder with the command:

```
MENUBUILDER >Listing3.c  Listing2.menu
```

The resulting menu code will be in the file "Listing3.c" for this example. The output from the menubuilder is collected using the standard redirection to the file. This makes the menubuilder code simple.

The bottom line is that you cannot use "its too complex or confusing" as an excuse anymore! With the MenuBuilder, you can create complex menus with lots of items and sub-items!

Once you start using menus, you might take a look at the intuition included files supplied with your C compiler. There are many of options not mentioned here. One of these options is Mutual Exclusion. Mutual Exclusion is a way to force one menu item to exclude others. The normal usage would be to allow a user to select only one printer drive out of many. Each printer driver mutually excludes all others. The MenuBuilder sets the Mutual Exclusion field to zero, allowing any or all items to be selected. You have to experiment to learn! The MenuBuilder will support only part of the options used in a program. By editing the result from the MenuBuilder, you can add your own options and still save alot of time creating menus.

The MenuBuilder is not perfect. It does not check for limits like do your menus fit on the screen. Do you have too many items, or sub-items? You will find that a menu description will compile and link properly only to see that it is messed up when you try to use it. That is the reason for the driver program (Listing One). With the Driver, you can create the "look" you want. You do not have to worry about testing everything at once. You can actually build up a program based on the driver! With these programs, you can actually create a menu, compile and link it with the driver, and have a prototype of your application running in just a few minutes.

The Second half of this article will describe in detail the actual code in the Menubuilder. The first thing to understand about the MenuBuilder is my coding style. I have seen many of articles about "Amiga-tizing" your programs. Well, I prefer the "if it works, don't mess with it" style of programming (remember from the first article, I'm the lazy one!). I like to use the print function for most of the output, even if it is just one character! In addition, I make all my defined parameters all capitals (it makes them easy to spot). Another characteristic of my programs is that since I use the CLI most of the time, I don't usually add the logic to support running them from the workbench.

Also important to note is that I use the Lattice compiler for all my work. I did not intentionally use any features that would stop this code from working with other compilers. If you have a problem or a question send me a note, and I will try to help to resolve it.

Let's start with the main program. It will use the command line arguments as filenames of the menu descriptions. Normally, one text file will be more than enough to describe your menus. I allowed for multiple input files just to make it easy to combine two or more menus into one program! If you combine menus, be aware that while duplicate menus and items names are allowed; it can cause confusion for the user.

The Main program has the job of:

1) Initializing the process

2) Opening each file

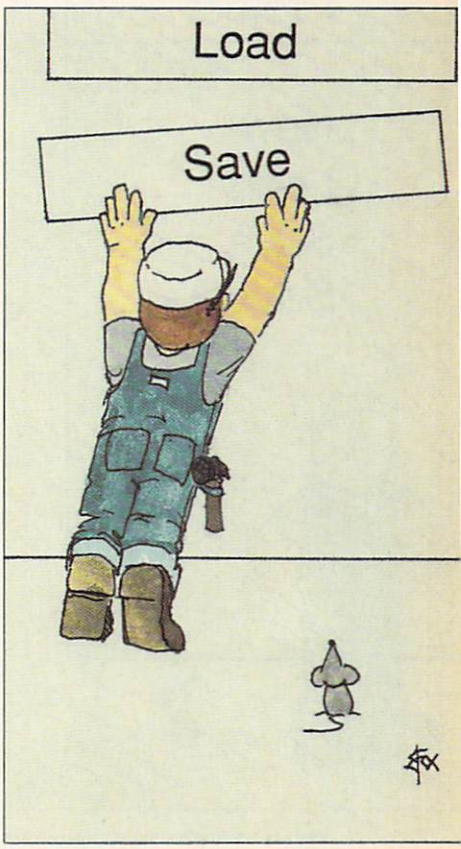
3) Processing the commands

4) Closing each file

5) Generating the resulting code

There are two functions that do most of the work for the main process. The first is Scan. Scan is a useful function that, given a pointer to a string, will either search for a character or the first occurrence of something that is not that character. This is a case where the code is shorter than the description! Scan will always point to a character in the string or at the end of the string (normally a string ends with a zero byte). Scan stops when it sees the zero byte or a newline character. Scan is a useful routine. I use it frequently when searching strings.

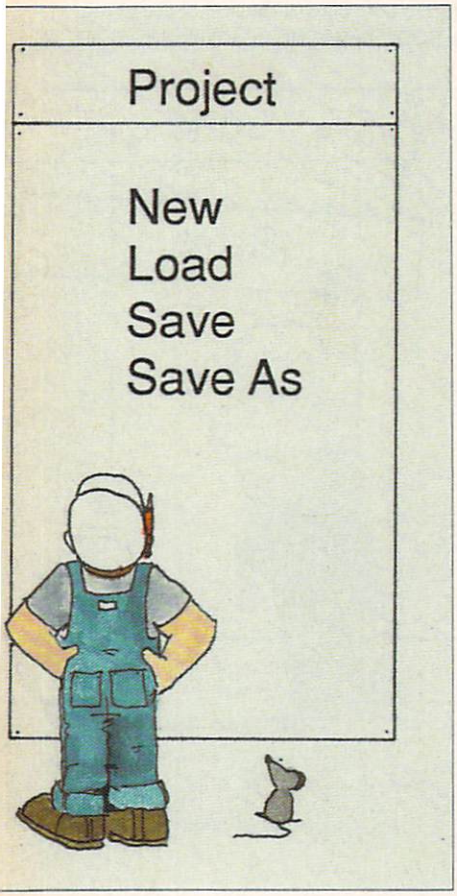
The second function is "strncmp". This routine is a "compare two strings of fixed length" function. It is a library function supplied by Lattice. I use it to decode the commands (MENU, ITEM, and SUBI). When "strncmp" finds a match, the string is sent to Scan to move to the comma following the command. When the command is MENU, ProcessMenu is called to take care of the rest of the string. If the command was ITEM or SUBI, ProcessItem is called to build the Item information.



When I designed this program, I decided to use the actual Intuition Menu and MenuItem structures to store the data needed to create the resulting code. In ProcessMenu, a Menu structure is built for each MENU command. This is done by dynamically allocating a Menu structure. There are two ways to implement a program of this type. The first is to create an array large enough to hold what you think will be the maximum amount of data. This will work most of the time, but it has a couple of drawbacks: Large arrays take up memory. It wouldn't matter how much data you have. If you are tight for memory, you might not be able to run the program! Another reason is that while you "think" you know what is needed to process the maximum amount of data, you may be wrong! I work for a company that has redelivered programs many times just to increase the sizes of some tables that could not be dynamically allocated. By using dynamically allocated memory for the structures, both of the problems go away! The program is smaller and we only use the amount of memory we really need. Everything in this program is made into linked lists allocated dynamically. The Menu structure is used to store the menu name, a pointer to the Items on the menu, and the length of the menu name. There are other fields in the structure, but I don't use them. Menus are linked together much like the actual menus. By studying how this program processes and stores the data, you will learn much about link list handling, not just menus. DoOutPut will travel these structures and create the code. Normally, when you are done, all allocated stuctures are released so that other programs can reuse the memory. Since I am a lazy programmer, I let the Lattice functions clean up after themselves! With the standard library routines malloc and calloc, Lattice keeps track of all allocated memory. When your program exits, a clean up routine will return the memory you used. For a simple low memory usage program like the MenuBuilder, this technique is fine. It saves a little work on my part. I normally try to deallocate each block of memory after it

is used. If you do not take care of this (or know that the compiler does for you), you will find memory disappearing from the system each time you run a program. Eventually, you will have to reboot the system to get the memory back.

When the ITEM command is processed, several error211 checks are made in the function ProcessItem. The first is that you must have a MENU command before an ITEM command! The second check is for the SUBI command. I use the ProcessItem function for both the ITEM and SUBI commands. There is a very good reason for this dual purpose! Both a Menu Item and a Menu sub-Item are defined by the same structure.The MenuItem structure. The way this works is that a Menu structure always points to a MenuItem structure using the FirstItem pointer. The Items that have sub-Items have a pointer to the same kind of structure (MenuItem structure) with the SubItem pointer. Both kinds use the NextItem pointer to create a linked list of either ITEMs or SUBIs (items or sub-Items) from your input. Each block represents either a Menu or MenuItem structure. The NULL is the indication of the end of the list. Your program reference the menu using the pointer MyMenu.



Menus have several complexities that can confuse even those familiar with them! One of the complexities here is the Menu header. You include the header text in the Menu structure directly. For Items and sub-Items, you can't just include the text. You must build another structure called an IntuiText structure and include a pointer to the MenuItem structure. The reason for this is to tell Intuition not only where to put pixel positions, but you also need information on the font (the default is used here) and what pens to use. Since I am creating simple text fonts, I chose some reasonable defaults. The text ends up being a foreground color on the regular background. I also use the default font. This turns most of the IntuiText structure into a bunch of constants. The IntuiText structure will be the first thing the output routine creates. The text for the Items and sub-Items will be saved in these structures by the function ProcessText. This function has the same type of "Create the List if new, or just add to the end" logic as the other functions. The one difference is that a search is done to eliminate duplicate Item names. If you use the name "DELETE" ten times in your Items, it will only have one structure and all ten Items will point to it! This is the one time that I freed up the memory. This guy returns the pointer to the IntuiText structure it either found or created. The pointer is saved in the MenuItem currently being built. The MenuBuilder repeats the process for the Item (or sub-Item) name and the Select name if you have one. Some times it is easier to see what happens than to explain what is happening! When a menu is dropped, the "Name" is displayed for each Item. When the mouse passes over the Item's position, Intuition will switch to the Selected name. This gives the menu a dynamic appearance. The last part of an Item or sub-Item is the command character. The command character is the character you use with the right-Amiga key as a menu equivalent for the keyboard. The maximum width of the menu item is kept in the first item's MutualExclude field. This will be used to size the menu.

When all the files are processed, a check is made to see if any errors occurred. If there are errors, the output routine is skipped. The output routine creates all the menu code. The code is generated in the order it is used. The IntuiText structures are created first just because every Item and subItem will reference them. The sub-Items are created next because all the Items will reference them. The menus are created last because they reference all the Items.

Creating the IntuiText structures is fairly simple, most of the entries are constants. You might try using JAM1 instead of JAM2 in the structures used with the Selected names included. Once you have a menu structure, you can experiment by substitution. (You can look at the Intuition Manual and the included file Intuition.h) for more information on the options available.

The sub-Items are created by traveling the menu list, and then a nested trip thru the Item list. When a sub-Item is found the DoOutSub function is called to generate the text.

The Items are done much like the sub-Items. The menu list is traversed, and each Item is generated. If the Item has a sub-Item, the proper pointer is created to the previously generated sub-Item structure.

The Menus are similar with the only difference being the Menu structure does not use the IntuiText structure for the heading text. The text is included in the actual menu structure itself.

Finally, the last code created is the pointer to the menu itself. This is the one that the test driver will reference as external. This is our connection to the driver. Your program will reference this external item when attaching the menu to your window.

When you look at the actual text generated in Listing Three, just after the IntuiText structures, there are a couple of preprocessor macros. A preprocessor macro is an aid to typing and readability. The "FLAGS" macro defines the Item and sub-Item characteristics. I selected the options that would:

1) CHECKIT - Mark a selected item with the standard check mark.

2) ITEMTEXT - Tell Intuition that we are using text not graphics.

3) MENUTOGGLE - Let you toggle an item and its check mark.

4) ITEMENABLED - Makes all items enabled initially.

The other define is just a short cut to select the right IntuiText structure and to keep the resulting text a little more readable. It allows the text for an Item to fit in less than 80 columns!

All the numbers in the Menu and MenuItem structures are calculated using the character width of eight pixels. This is OK for our default font and the normal menu, but if you wish to use larger fonts you will need to adjust the pixel positions. As mentioned previously, the positions of the menus and the Items are calculated. Each Menu is given a width of the heading length plus 2 characters wide. This gives a readable heading display. The position of any

(continued on page 74)

No Fishing!



The Ones That Got Away.

A Closer Look at PD Software Not Currently in the Fred Fish Collection.

by Graham Kinsey

This month's highlights include a stunning preview of what Workbench 1.4 will be like, plus a fantastic gambling simulation. For each program, the author's name is given when it is known, and in most cases, the AmigaZone download file number will be listed as well (this doesn't mean that if a file number isn't given that it is not on PeopleLink at the moment, simply that I obtained it from somewhere else) so those of you who do have access to PeopleLink and its AmigaZone can quickly download this file. When a public domain program has been classified as shareware, this is also mentioned with the suggested amount if one is given. Due to the large size of animations that are coming out for the Amiga, I have decided to assume that the normal size of an Amiga animation is one megabyte. This means unless I specify otherwise, ALL animations reviewed here require one megabyte in order to run. If you are an Amiga owner with only 512K in your machine you should keep this in mind until you can add some more memory to your Amiga.

TOOMUCH3D

TOOMuch3D: by Jim Robinson (AmigaZone file #19837). For those who have been watching Jim's other VideoScape 3-D masterpieces a little too much lately, this one's for you. While this is not nearly as long as his other animations, it is still quite good (especially the 3-D effect with the chair). Like MarsFlight this animation also begs for an accelerated Amiga to run.

AMIMODEM

AmiModem: by Mike Thomas (AmigaZone file #17223). For those who are trying to wade through the dozens of settings for the Courier HST 9600/14.4K baud modem, AmiModem can help. This program (written by the author of AmiExpress, a commercial BBS program designed especially for this popular modem) will set the NRAM settings for you.

ATOMCLOCK

AtomClock version 1.2: by Art Steinmetz (AmigaZone file #17741). This new version of AtomClock has one very important addition in the form of ARexx support. This now makes it possible for those who own both ARexx and WSHell (Bill Hawes' commercial shell) to have AtomClock automatically grab the time from the U.S. Naval Observatory in Washington once a month.

CARDCOMM

CardComm: by John Hoffman (AmigaZone file #18263). Yet another derivative of DJ James' Communicator. Like the now defunct (since it is no longer supported on PeopleLink) Graphical Teleconferencing program ACO, CardComm also can only be fully used on PeopleLink. CardComm's forte is in allowing users on PeopleLink to play all sorts of card games using a graphical interface within CardComm. Not only does it support card games with differing amounts of cards dealt (all the way up to 13), a sophisticated scoring system is also included in CardComm. CardComm also fully supports MoreRows displays, which Communicator 1.34 does not IF you have the 1.3 Kickstart ROMs installed in your Amiga (since CardComm looks exactly like Communicator 1.34 does except for the added card game features, some users of Communicator 1.34 may want to switch over to this variant of Comm 1.34).

HERMIT

Hermit: by A.G. Kartsatos (AmigaZone file #17905; Shareware: $15). Another screen saver utility whose unique ability is to be able to save the font used on any screen. This now allows you to grab a custom font that is buried inside an executable program from the program and save it to disk for your own use.

ICONMASTER

IconMaster: by John Scheib (AmigaZone file #17622; Shareware: $10). Without a doubt the most complete program for creating icons. IconMaster's paint screen includes such features as undo, dithered colors, a magnification window, AutoCenter, swapping of pen colors, and more. The main control screen of IconMaster is where you can edit the file protection bits of an icon, alter an icon's ToolTypes, change the stack size, and more. Harv Laser (Senior Chairperson of PeopleLink's AmigaZone) summed it up best when he referred to IconMaster as "DeluxePaint for Icons".

CHARPIC

CharPic: by Kelly Day (AmigaZone file #18097). A unique program that converts a monochrome 640x200 IFF picture into a set of ASCII characters. Not only does this make it easier to transport graphics via modem, it stands alone as an effect-generator (an IFF-format animation using CharPic-generated frames shows the effect off very nicely).

NOCLICK

NoClick: by David Bolt (AmigaZone file #19166). Finally, someone has eliminated a minor (at least in the eyes of Commodore) but still very annoying problem with the Amiga; the clicking of empty floppy drives! NoClick will indeed stop the clicking of floppy drives on ANY Amiga, including 500's and 2000's. NoClick has already been through a few revisions, and the current version, 3.6, will work on any Amiga even if it is not running with a 68000 (earlier versions of NoClick crashed on my 68010-based Amiga). NoClick will not stop the clicking on all floppy drives however (while my internal floppies on my 2000 are now silent, my ancient A1010 doesn't respond to NoClick), and it is possible for damage to occur on non-responding floppies in the long-term (always keeping a floppy disk in the problem drive should help to reduce the chance of damage occurring). Needless to say this is one utility all Amiga owners will want to get their hands on.



NOW FOR THE AMIGA!

Don't let those finger aerobics get you down! There's an easier way to change between your mouse and joystick or joystick and dongle without all that cable swapping. Mouse Master lets you instantly switch with just a touch of a button. It's housed in a compact case for stylish good looks and includes its own custom cable. It's the handiest switchbox around!

MOUSE MASTER™

Orders/Inquiries ☎ (602) 322-6100

ONLY **$39.95**
plus shipping & handling
MC, Visa, COD's welcome

Practical Solutions®

©1989 fax (602) 322-9271 • 1135 N. Jones Blvd. • Tucson, AZ 85716

Circle 137 on Reader Service card.

PATENT

Patent: by Jay Miner (AmigaZone file #17781). If you like to collect Amiga memorabilia this is worth getting. This text file and accompanying IFF pictures are a copy of the actual documents filed by Amiga Inc. in July 1985 to obtain a U.S. patent on the Amiga computer. While most of the text may be greek to those who know little about computer hardware, it is still fascinating to view.

KNIGHTERRANT

KnightErrant: by Eric Fleischer (AmigaZone file #18266). Another fine animation by Dr. Gandalf. This one depicts a knight in VERY shiny armor, who after walking in the desert for a long time, suddenly does his best "Amiga impression". There are actually two animations included in the WARPed files, the other one is a close-up shot of the "impression" called Knight-Closeup. KnightErrant was created with TurboSilver V3.0 (with the assistance of Diamond, and the sounds were created via PerfectSound). The KnightErrant will just barely run on a 512K Amiga (Knight-Closeup will not).

FRANKLIN FLY

Franklin Fly: by Avril Harrison. You don't have to purchase high powered and high priced animation software like TurboSilver and Sculpt-Animate 4D (nor learn how to program The Director) in order to produce great animations on the Amiga. Avril Harrison's "A Day in the Life of Franklin Fly" showcases the capabilities of one of the Amiga's low priced (and easy to use) animation packages, Fantavision. This long animation is comprised of several scenes from the typical life of a fly. In this animation Franklin Fly overcomes such obstacles as a spider's web, a window, and a big bowl of soup. In another scene the Queen of England makes a guest appearance that she would rather forget quickly! This animation isn't a showcase of the Amiga's graphics capabilities (though an MS-DOS machine would still be hard pressed to match it!), but it does include many great sound effects, is very funny at times, and overall is a pleasure to watch. Franklin Fly also requires just 512K in order to run. Despite being a low-budget production, Siskel & Ebert would still give Franklin Fly two big thumbs up.

DOCTORA

DoctorA: by Marvin Landis (AmigaZone file #17598). The third animation featuring that ultimate computer athlete, AmiGuy. A sequel to the BoingThrows animation, DoctorA features AmiGuy showing off some serious hang-time on the court. While the animation segment isn't as long as in Gymnast nor BoingThrows, DoctorA is still as stunning as the other two. DoctorA was also created with Sculpt-Animate 4D, and really needs one megabyte in order to run (Marvin does explain how you can run a crippled version of DoctorA with just 512K).

JAZZBENCH

JazzBench: by David Navas (AmigaZone file #18012). If you just can't wait for Commodore to release Workbench 1.4, and you don't have developer status (and therefore can't get your hands on pre-release versions of 1.4), now you finally can get a feel for what Workbench 1.4 will be like in many ways, thanks to JazzBench! JazzBench is a TOTAL replacement for the Workbench environment, and as you might guess, it is wildly different! First of all, you can display AmigaDOS Dir-style text listings in a JazzBench window. JazzBench windows open in a Macintosh-type fashion (i.e. they explode), and any window on the JazzBench screen can be iconized (which means any program that lives on the Workbench disk can be iconized!). JazzBench is also asynchronous just like Workbench 1.4 will be (you won't be seeing much of the busy pointer in Jazzbench). The menus for JazzBench are much larger, including menu items for many AmigaDOS commands, plus you can install menu items for your own applications. Icons can not only be sorted in different ways, but you can even have a default icon appear for programs that don't have their own icons. I've only scratched the surface here, JazzBench holds much more for those who use it. JazzBench is not designed for a 512K System, but the author does explain how you can run a limited version of JazzBench on a 512K Amiga. While the current version of JazzBench is not stable enough for long-term use as the default environment (neither is Workbench 1.4 at this point), it is still a must-see for those who can't get their hands on Workbench 1.4.

PROTECTA

PROTeCTA: by A.G. Kartsatos (AmigaZone file #17785). A simple and nicely-done gadget-laden program for changing

the protection bits of a file. This program will not, unfortunately, work from the Workbench (since the author would naturally prefer Workbench users to check out his full-featured DirUtil program DISKoLAB).

SUBLIMINAL

Subliminal: by Steve Tibbett (AmigaZone file #17971). Another unique program, this will indeed allow you to create and display subliminal messages on your Amiga. Subliminal will display up to 20 messages once every 5 minutes. Subliminal does this on the Workbench screen (making it ideal to use on unsuspecting subjects who tend to work long hours in the CLI or in window-based wordprocessors like WordPerfect and Scribble).

TITLEGEN

TitleGen: by Kevin Kelm (AmigaZone file #18192; Shareware: $10). A very simple but useful video titling program. This program can only scroll text from bottom to top and doesn't support any wipes (except for just popping in a screen of text). All text and effects are created in an external text file, so it is possible to type something up with a word processor, save it as ASCII and then have TitleGen display it. TitleGen supports ten scrolling speeds, and supports a few basic options including changing the colors (only three can be used at one time), time delays, halting the display until the LMB is pressed, and a loop option. Of course TitleGen will accept any font, and centers all text. This is the only decent video titling program I've seen in the public domain (unless you would label Startle as a video titling program) so send in those shareware checks if you want to see something better. The author did state that he is considering improving this program if enough people care.

BMP

BMP version 2.0: by J.L. White ((AmigaZone file #18396; Shareware: $10). Version 2.0 of this great IFF sound player. BMP no longer is limited to using CHIP RAM when playing sound samples, so you can now play multi-megabyte sample with it (including using it to play samples created with AudioMaster II). BMP also now includes an ARP-style file requester (and as a result you now must have the arp.library in your LIBS: directory to use BMP).



Circle 134 on Reader Service card.

ESHFIX

EshFix: by Tom Eshelman (AmigaZone file #18152). AmigaDOS's Ed editor can now load in many more text files than it used to be able to thanks to EshFix. Eshfix will strip any and all binary characters from a text file so that it will load in to Ed. EshFix doesn't alter the original file, it only creates an altered version (be aware of that if your storage device is almost full).

HPGL2PS AND XDRAW

HPGL2PS and XDraw: by Rudolf Werner (AmigaZone file #'s 19144 and 19143). Two essential utilities for users of XCad. HPGL2PS will convert the HPGL plotter output from XCad into PostScript format. XDraw allows you to import spreadsheet data from MaxiPlan III/PlanIt/MaxiPlan (pick your name) or other spreadsheet programs into XCad. This makes it possible to import coordinate data into XCad without having to typing it into XCad directly.

MKDDIR

Mkddir version 1.2: by Ron Shaw (AmigaZone file #18393). Version 1.2 of the ONLY CLI-based directory creation command that really supports Workbench. Mkddir's newest release supports multiple drawer icons types, including a couple of 3-D drawer icons (which can be selected either via ARP's Set command or via the included set utility in Mkddir). A very handy program for those who create PD disk collections or other types of Workbench-supporting software.

PPLANE

PPlanes: by Michael Fahrion (AmigaZone file #18196). The old pastime of making paper airplanes has now been greatly enhanced by the Amiga! PPlanes includes IFF pictures to help you create your own multi-color paper airplanes with any paint program that supports 640x400 resolution (course you could just resize the pictures if you want your own HAM air force instead). In addition to the monochrome template and folding directions, there is also a finished design and some very nicely rendered paper airplane clip-art to quickly jazz up your own designs (US markings only in this set).

QUICPREF

QuicPref: by David Junod (AmigaZone file #18195). If you print graphics often this utility may find a home in your system. QuicPref duplicates the Graphics 2 screen in Preferences (in about one tenth

Now In Wisconsin! Order Toll Free 800-558-0003

Since 1982 ComputAbility Consumer Electronics

Now In Wisconsin! Order Toll Free 800-558-0003

AMZ 1-90

AMIGA SOFTWARE

688 Attack Sub	31.95	Contra	24.95	Firestone	21.95	Karafonts Subhead	44.95
A-Talk 3	64.95	Cribbage King/Gin King	37.95	Flash Dragon	21.95	Keef the Thief	31.95
Aaargh	24.95	Cross DOS	18.95	Flight Simulator II	30.95	Killing Game Show	21.95
Action Fighter	24.95	Curse of the Azure Bonds	26.95	European Scenery Dsk.	19.95	Kind Words V2.0	57.95
Advantage, The	CALL	Dark Castle	28.95	Focus Video	107.95	King's Quest 1,2,3 or 4	30.95
After Burner	30.95	Dark Century	28.95	Forgotten Worlds	24.95	Kingdoms of England	26.95
Alien Syndrome	24.95	Data Storm	25.95	Future Sound 500	89.95	Knight Force	28.95
AmigaDOS ToolBox	38.95	Deathbringer	25.95	Gauntlet II	30.95	Knights of Legend	30.95
Animagic	58.95	Defender Of The Crown	31.95	GFA Basic 3.0	87.95	Kristal, The	31.95
Anim 4-D Jr.	87.95	Deja Vu 1 or 2	30.95	Gold of the Realm	24.95	Lancelot	31.95
Aqua Blast	23.95	Deluxe Music Constr 2.0	61.95	Gold Rush	24.95	Last Duel	24.95
Aquanaut	25.95	Deluxe Paint III	97.95	Gore	24.95	Last Inca, The	24.95
Archipelagos	23.95	Deluxe PhotoLab	97.95	Grabbit	18.95	Lattice C	199.95
Arexx	28.95	Deluxe Print II	51.95	Grand Prix Circuit	27.95	Leisure Suit Larry	25.95
Aunt Artic Adven.	24.95	Deluxe Productions	128.95	Gunship	34.95	Leisure Suit Larry II	30.95
Artura	25.95	Deluxe Video III	94.95	Halls of Montezuma	28.95	Leisure Suit Larry III	37.95
Awesome Arcade Pack	30.95	Deluxe Video V1.2	84.95	Hardball II	28.95	License to Kill	21.95
B.A.D.	29.95	Denaris	24.95	Harrier Combat Simulator	30.95	Life and Death	31.95
Balance of Power 1990	30.95	Design 3D	59.95	Heat Wave	28.95	Logicworks	184.95
Ballistyx	21.95	Designasaurus	30.95	Heavy Barrel	24.95	Lords of the Rising Sun	31.95
Bar Games	28.95	Digidroid	59.95	Heavy Metal	28.95	Luxor	19.95
Bars and Pipes	179.95	Digipaint3.0	84.95	Heros of the Lance	25.95	Magellan 1.1	118.95
Battle Chess	29.95	Digiview Gold	124.95	Heros Quest	37.95	Magic Johnson 512	24.95
Battle Hawks 1942	31.95	Digiwork 3D	82.95	Hillsfar	31.95	Magic Johnson 1 Meg	31.95
Battle Squadron	24.95	Disk 2 Disk	28.95	Hole In One	21.95	Manhunter - New York	30.95
Battles of Napoleon	31.95	Disk Master	28.95	Home Builder's Friend	68.95	Manhunter-SanFrancisco	30.95
Battletech:The Cr Hk Inc.	30.95	Disk Mechanic	54.95	Hostage	29.95	Maniac Mansion	28.95
Baud Bandit	30.95	Distant Sun	30.95	Hound of Shadow	25.95	Matrix Marauders	21.95
Berzerk 3	22.95	Dos 2 Dos	30.95	Hoyles Book of Games	21.95	Mavis Beacon Typing	31.95
Black Cauldron	24.95	Double Dragon	24.95	Hunt For Red October	31.95	Menace	19.95
Blasteroids	CALL	Double Dragon II	24.95	Ikari Warriors	24.95	Miami Vice	24.95
Block Out	24.95	Doug's Math Aquarium	46.95	Indiana Jones-Action	25.95	Michton Hit Disk #1	30.95
Blood Money	25.95	Dr. Doom's Revenge	28.95	Indiana Jones-Graphic	30.95	Micro Fiche Filer Plus	114.95
Blue Angels	28.95	Dragon Force	31.95	Infestation	24.95	Midi Rec Studio V1.1	39.95
Boardmaster	57.95	Dragon's Lair	33.95	Interchange	28.95	Might and Magic II	37.95
Boomerald	24.95	Dragon's Lair II	CALL	IntroCad	46.95	Modeler 3D	56.95
Breech II	30.95	Dragons of Flame	25.95	Invision	134.95	Monopoly	24.95
Bride of the Robot	24.95	D.U.D.E.	29.95	Innvision +	174.95	Movie Setter	59.95
Broadcast Titler	184.95	Dungeon Master	24.95	It Came From The Desert	31.95	Music Studio 2.0	49.95
Can Do	84.95	Dungeon Master Asst 2	29.95	Jack Nicklaus Golf	30.95	Music X	169.95
Carmen SanDiego/ USA	30.95	Dungeon Quest	31.95	JN Courses of 1989	14.95	Netherworld	22.95
Carmen SanDiego/ World	28.95	Dunlap Utilities	47.95	JN Course Disc II	14.95	Neuromancer	28.95
Carthage	24.95	Earl Weaver Baseball	31.95	Joan of Arc	28.95	Never Mind	21.95
Chaos	24.95	Elan Performer	35.95	Journey	31.95	New York Warriors-1 Meg	30.95
Chessmaster 2000	28.95	Excellence 1 Meg	158.95	Kampfgruppe	37.95	Night Dawn	18.95
Chessmaster 2100	31.95	Exolon	22.95	Karafonts Headlines 1 or 2	44.95	Night Force	28.95
Chronoquest	30.95	Extend 1.3	23.95				
City Desk 2.0	112.95	Eye of Horus	24.95				
Clue Master Detective	24.95	F-16 Combat Pilot	31.95				
Codename: Iceman	37.95	F-40 Pursuit Sim.	28.95				
Colonel's Bequest	37.95	FA/18 Interceptor	31.95				
Colony	30.95	Falcon	31.95				
Combat Course	25.95	Operation Counterstrike	16.95				
Comic Setter	59.95	Fantavision	37.95				
Comic Art Disks (each)	20.95	Fast Track	25.95				
Conquest of Camelot	37.95	Fiendish Freddy	30.95				

Nimbus	97.95	Pub Games	25.95	Superbase Pro 3.0	189.95		
Nuclear War	30.95	Puzzle Storybook	25.95	Superplan	87.95		
Omega	30.95	Quick Silver	41.95	Superman	24.95		
Omniplay Baseball: 5 on 5	30.95	Qix	22.95	Superstar Ice Hockey	30.95		
Omniplay Horse Racing	30.95	Quarter Back	42.95	Stunt Car	25.95		
On Line Platinum	58.95	Quasar Sound	34.95	Sword of Sodan	28.95		
Operation Wolf	22.95	Questron II	31.95	Swords of Twilight	31.95		
Optics	117.95	Rambo III	22.95	Targhan	24.95		
Oswald	23.95	Rampage	24.95	Teenage Ninja Turtles	28.95		
Outrun	30.95	Rastan	22.95	Temple of Doom	31.95		
Over Run	32.95	Raw Copy	34.95	Tetris	20.95		
Page Renderer 3-D	94.95	Red Lightning	37.95	Text Ed Plus	48.95		
Page Setter	59.95	Renegade	22.95	Test Drive 2 - The Duel	27.95		
Page Stream	127.95	Rhyming Notebook	25.95	California Chall.	14.95		
Dot Matrix Font1-5	ea. 24.95	Rick Dangerous	21.95	European Chall.	14.95		
Page Thinker	59.95	Rick Davis Soccer 1 Meg	30.95	Muscle Cars	14.95		
Pagefliipper Plus/FX	87.95	Rick Davis Soccer 512K	24.95	Supercars	14.95		
Panzer Strike	29.95	Ring Side	24.95	Their Finest Hour	37.95		
Paperboy	30.95	Risk	24.95	Thinker	57.95		
Pen Pal	84.95	Road Challenge	24.95	Third Courier	28.95		
Phasar	54.95	Robocop	28.95	Thud Ridge	29.95		
Photon Paint 2.0	87.95	Rocket Ranger	31.95	Thunder Blade	29.95		
Pic Magic	57.95	Roger Rabbit	28.95	TKO	24.95		
Pipe Dreams	25.95	Romance/Three Kingdoms	49.95	Trackers Quest	21.95		
Pbal Script	92.95	RVF Honda	25.95	Turbo Silver 3.0 1 Meg	112.95		
Planet of Lust	24.95	Savage	21.95	TV Show	64.95		
Police Quest 1 or 2	30.95	Schufflepuck Cafe	24.95	TV Text	64.95		
Pool of Radiance	31.95	Scrabble	24.95	TV Text Professional	99.95		
Populous	32.95	Scribble Platinum	87.95	Twilight's Ransom	19.95		
Postscript Fonts A,B,C	ea. 24.95	Sex Vixens-Outer Space	22.95	Typhoon of Steel	31.95		
Power Windows 2.5	54.95	Shadow of the Beast	30.95	Ultima 4	38.95		
Powerdrome	25.95	Shark Attack	24.95	Ultima III	24.95		
Precisely	48.95	Shiloh	25.95	Universe 3	31.95		
Print Master Plus	24.95	Shinobi	30.95	Veteran	19.95		
Pro Script	28.95	Shufflepuck Cafe	24.95	Video Effects 3-D	113.95		
Pro Soccer	21.95	Slideshow	27.95	Video Fonts	28.95		
Pro Sound Design	96.95	Sim City	28.95	VideoScape 3D 2.0	115.95		
Pro Video Gold	174.95	Skate Wars	31.95	Vindicators	CALL		
Pro Write 2.5	89.95	Sky Shark	22.95	VIP Professional	59.95		
Pro-Draw	112.95	Skyline 1.2	86.95	Vista 3-D	87.95		
Professional Page	167.95	Solitaire Royale	19.95	Viva-Amiga	122.95		
Structured Clip Art	36.95	Sorcerian	30.95	VTX Online	51.95		
Templates	36.95	Sorceror Lord	22.95	W Shell	28.95		
Profonts 1 & 2	21.95	Space Quest 1 or 2	30.95	War Game Construction	19.95		
Project D+ Backup	30.95	Space Quest 3	37.95	War of the Lance	25.95		
Prospector Mazes of Zor	24.95	Space Racer	19.95	Wayne Gretzky Hockey	30.95		
		Speedball	25.95	Wild Streets	28.95		
		Star Command	31.95	Willow	25.95		
		Star Trek: The Final Front.	34.95	Wind Walker	25.95		
		Star Wars	24.95	Wings Of Fury	25.95		
		Stellar Crusade	36.95	Wizard's Crown	25.95		
		Strip Poker II	23.95	Works Platinum, The	CALL		
		Stryx	21.95	World Atlas	35.95		
		Super Back	48.95	X-Specs	84.95		
		Superbase Personal	46.95	Zak McKracken	28.95		
		Superbase Personal 2	87.95	Zork Zero	38.95		

MISCELLANEOUS HARDWARE

AMI GEN	$99	CHINON FB 354 INT. DR.	$82
SUPER GEN	$669	COLOR SPLITTER	$119
SUPER GEN 2000S INT GEN	$1449	GOLD DISK SCANNER	$899
MAGNI 4004 GEN LOCK	$1389	IMG SCAN	$105
FLICKER FIXER	$455	SCAN LOCK	$819
PERFECT SOUND	$69	GVP 68030 ACC. BD	CALL

IVS Trumpcard Hard Drive Packages for A2000 Series

Seagate

ST-157N-1 49 MEG.-28ms 519 DEL
ST-177N 60 MEG NEW! 679 DEL
ST-277N-1 60 MEG 619 DEL
ST-296N 80 MEG 629 DEL

Quantum Pro Drive

40 MEG SCSI Pro Drive 619 DEL
80 MEG SCSI Pro Drive 869 DEL
100 MEG SCSI Pro Drive 949 DEL

these kits include IVS Trumpcard 8C81 hard drive ctrl, cable, software and **FREE delivery** in the contiguous USA This is not anassembly kit ! It is a package.

Hard Drive Cards (A-2000)

Seagate

ST-157N-1 49 MEG - 28 ms 539 DEL
ST-177N 60 MEG NEW ! 699 DEL

Quantum Pro Drive

40 MEG SCSI Pro Drive 639 DEL
80 MEG SCSI Pro Drive 889 DEL
100 MEG SCSI Pro Drive 969 DEL

these cards include IVS Trumpcard ctrlr, mounting brkt, cbl, software, and **FREE delivery** in the cont-iguous USA.

MEMORY

HURRICANE

Hurricane 2800/68030 Call
Hurricane 500/68020 Call
Pro Ram 1.8/512k 119 Del

SUPRA

Supra Ram 500 93 Del
Supra Ram 2000 - 2 MB 309 Del
Supra Ram 2000 - 4 MB 469 Del
Supra Ram 2000 - 6 MB 629 Del
Supra Ram 2000 - 8 MB 789 Del

Includes**Free** delivery in the Continental USA

Trumpcard 500 Hard Drive Packages for Amiga 500

Seagate

ST-157N-1 49 Meg -28ms..579 Delivered
ST-177N-60 Meg NEW ...729 Delivered!

Quantum Pro Drive

40 Meg SCSI Pro Drive - 679 DELIVERED
80 Meg SCSI Pro Drive - 929 DELIVERED
100 Meg SCSI Pro Drive - 999 DELIVERED

These Packages include Trumpcard 500 enclosure, IVS Trumpcard ,SCSI hd drive Ctrl, Software & free delivery in the contiguous USA

Trumpcard 500 Ram Expander available- 512k • 1 MEG • 2 MEG • 4 MEG

Seagate SUPER SALE

ST-125N 20 Meg - SCSI 269
ST-138N 30 Meg - SCSI 299
ST-157N 49 Meg - SCSI 329
ST-157N-1 49 Meg - 28 ms 349
ST-177N 60 Meg - SCSI 469
ST-225N 20 Meg - SCSI 255
ST-251N 40 Meg - SCSI 375
ST-277N-1 60 Meg - SCSI 429
ST-296N 80 Meg - SCSI 459

Quantum

Pro Drives

40 Meg SCSI Pro Drive.....449
80 Meg SCSI Pro Drive.....699
100 Meg SCSI Pro Drive.....779

XETEC Fast Card

HardDrive Packages for Amiga 2000

Seagate

ST-157N-1 49 Meg-28ms 519 Del
ST-177N-60 Meg NEW! 679 Del

Quantum Pro Drive

40 Meg SCSI Pro Drive 619 DEL
80 Meg SCSI Pro Drive 869 DEL
100 Meg SCSI Pro Drive 949 DEL

Card features Xetec's DMAx hard, autoboot ROM,auto-config circuitry..25 pin SCSI conn., disk, utilities manual.

FREE DELIVERY IN THE CONTIGUOUS 48 STATES

XETEC *Fast Card System*

Hard Drive Packages for Amiga 500

Seagate

ST-157N-1 49 Meg-28ms 699 Del
ST-177N-60 Meg NEW! 849 Del

Quantum Pro Drive

40 Meg SCSI Pro Drive 799 DEL
80 Meg SCSI Pro Drive 1049 DEL
100 Meg SCSI Pro Drive 1129 DEL

System includesAdaptor with autoboot ROM, enclosure, manual, Fast Trak disk with utilities & shielded SCSI cable. With power supply and fan

FREE DELIVERY IN THE CONTIGUOUS 48 STATES

CALL US FOR ALL YOUR COMMODORE AMIGA™ NEEDS

VIDEO PACKAGE

PANASONIC 1410 CAMERA
16MM LENS WITH VARIABLE IRIS
COPYSTAND WITH LIGHTS
DIGIVIEW GOLD
$419 DELIVERED!

MASTER 3A Disk Drive

1010 Compatible $125

*Free Delivery to the Contiguous States

PRINTERS

Panasonic 1180 185
Panasonic 1191 235
Panasonic 1124 285
Panasonic 1624 NEW 429
Star NX-1000 159
Star NX-1000 Multi Font 2 NEW NEW NEW 169
Star NX-1000 Rainbow (color) 219
Star NX-2400 275

Modems

Avatex 2400 HC 129
Avatex 2400E 109
Avatex 9600 External NEW! 839
Supra 2400zi Internal (A2000) 139

Supra 2400 Ext. Modem & Cable

• $119 •

FREE DELIVERY to the 48 Contiguous States

VISA MasterCard

NO CREDIT CARD SURCHARGE

ORDER INFORMATION
414-357-8181 FAX 414-357-7814
P.O. BOX 17882 Milwaukee, WI 53217

ORDERING INFORMATION: Specify system. For fast delivery send cashier's check or money order. Personal and company checks allow 14 business days to clear. School P.O.'s welcome. C.O.D. charges are $4.00. In Continental U.S.A. include $3.00 for software orders 5% shipping for hardware, minimum $6.00. MasterCard and Visa orders please include card #, expiration date and signature. WI residents please include 5% sales tax. HI, AK, FPO, APO,Puerto Rico and Canadian orders, please add 8% shipping, minimum $4.00. All other foreign orders add 15% shipping, minimum $15.00. All orders shipped outside the Continental U.S.A. are shipped first class insured U.S. mail. If foreign shipping charges exceed the minimum amount, you will be charged the additional amount All goods are new and include factory warranty. We do not guarantee compatibility & version #s. Due to our low prices all sales are final. All Defective returns must have a return authorization number. Please call (414) 357-8181 to obtain an R.A. # or your return will not be accepted. Prices and availability subject to change without notice. Shipping and handling are non-refundable. We ship the latest versions available to us, updates must be handled by end user directly with the manufacturer.

Circle 117 on Reader Service card.

the size) and lets you change any of these settings. QuicPref can even pop in front of any screen for convenience, IF you are using version IV of PopCLI.

CRAPS

Craps: by Art Tedeschi (Shareware: $20). If you care at all about the game of Craps you'll love this simulation. Las Vegas Craps supports the Amiga's graphics, animation, and sound capabilities about as well as you can expect for a shareware program (including the HUGE icon for the main program). Of course this simulation allows every bet that you would be able to make in a Las Vegas casino, plus you can set the minimum bet and the odds multiple. If you know nothing about Craps, don't worry since this program makes learning easy and fun. In addition to a very nice HELP system which can explain more about the game than you would ever want to know, there is also a "Craps Assistant" which will tell you which bets are wise to make. Even more impressive is the Strategy editor/simulator where you can create betting profiles of a person by recording everything they do (or just input a betting profile on your own) and then test the soundness of it by simulating an actual Craps game environment with up to four simulated betters. There are other nice features like a percentage breakdown of all rolls that have been made (including comparing the number of times a dice number has come up with the number of times it should have appeared), animated dice and stick and true random numbers (based on the number of microseconds the Amiga has been on). Serious competition for MicroIllusions' commercial craps simulation Craps Academy.

HOCKEY

Hockey version 1.2: by John Samuels (AmigaZone file #18712). An update to the almost forgotten abusive card game Hockey. New features include a new statistic called "The Human Stupidity Factor", which shows how many times your imperfect logic has caused you to play the wrong card. You can also record a play-by-play log of a game, and the Amiga's card playing ability is now even better (not that it needed any help, but you know that your Amiga hates to lose, right?). Finally, you can edit the insults, file the Amiga uses, and input your own just in case you want the Amiga to humiliate and anger its human opponent even more.



Circle 123 on Reader Service card.

SETMEM

SetMem: by Mike Berro (AmigaZone file #18610). For programmers and others who must find out how much memory you need to run a piece of software, SetMem allows you to specify (to as close as eight bytes) how much FAST and CHIP memory you want to be available in the system.

SHOWFONT

ShowFont version 4.0: by Arthur Johnson (AmigaZone file #18319). Version 4.0 of the font-viewing program ShowFont. The big addition here is that you can reassign the FONTS: directory from within the program (NOTE: ShowFont will NOT reassign the FONTS: directory when you exit the program). You can change the number of bitplanes used in ShowFont, not only to speed up scrolling display, but also as "preparation" for support for color fonts (according to the author).

SPEED

Speed: by Jez San (AmigaZone file #18464). Competition for the Cramden Utilities' SI program has finally arrvied from assembly master Jez San (author of the StarGlider). Speed will benchmark your Amiga's CPU/memory speed compared to a stock A2000/A2500 (in terms of ratios).

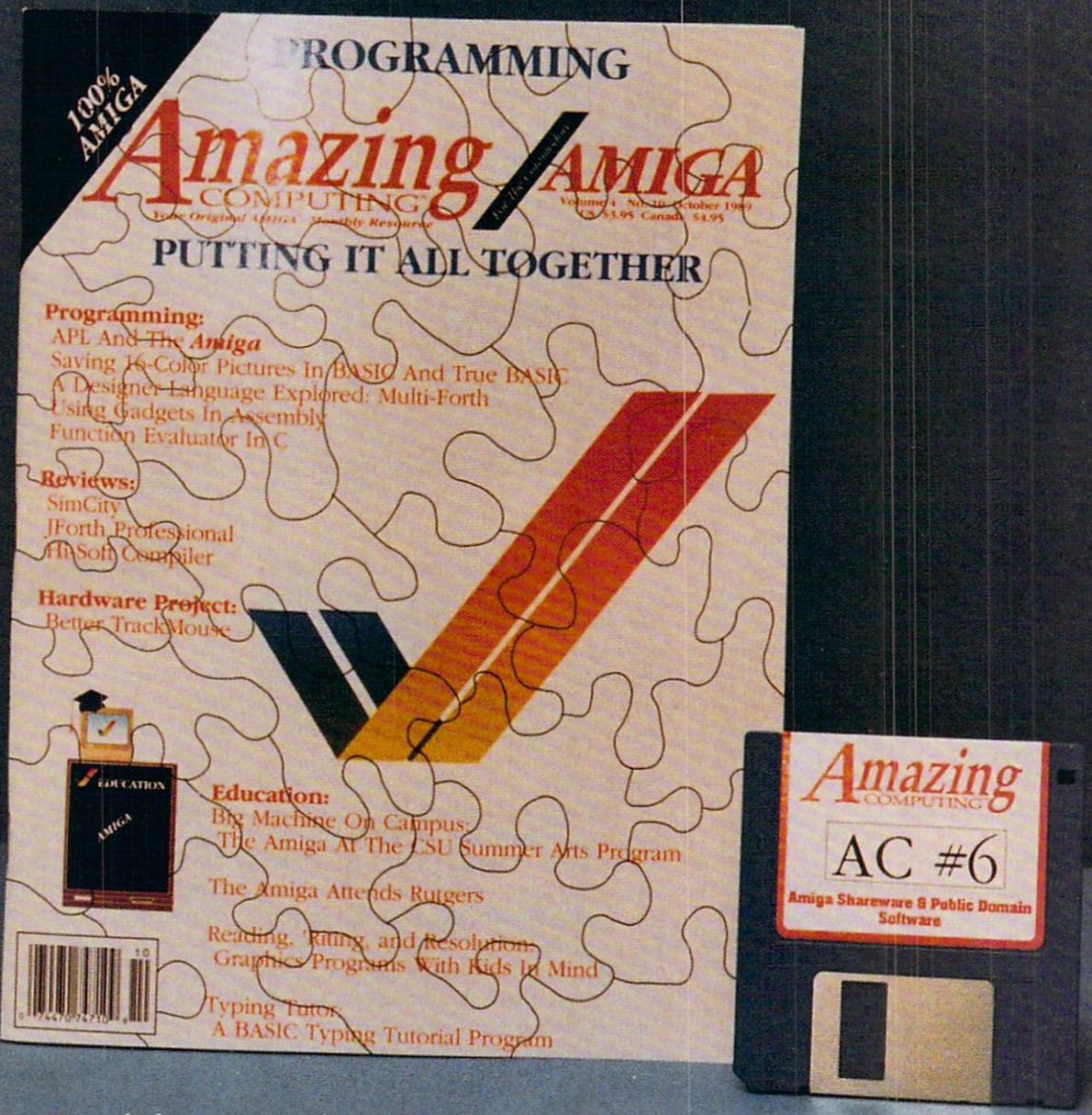
Next month, I will be talking about some great animations, including many from the often ignored commercial animation program Animator:Apprentice. As always, I can be reached on the AmigaZone on PeopleLink (ID: G KINSEY), or on the IDCMP BBS (617-769-3172, 3/12/2400 baud, 105 Megabytes online, running 24 hours a day, addressed to SYSOP). If you have written a public domain/shareware/freely distributable program, or have obtained one that you think is worth mentioning to all Amiga owners, then please attempt to contact me via the above contacts, or through *Amazing Computing*.

For information on obtaining some the programs that aren't listed as being on PeopleLink (or for those who don't have a modem), please write (and/or send $2 for an Amiga PD catalog disk) to:

SMAUG
1015 S. Artery #112
Quincy, MA 02169

•AC•

CHOICE!



Amazing Computing believes its readers should have a choice! AC provides a superior magazine, as well as a special disk with programs and background information, each month. Whenever possible, AC combines two issues on one disk and maintains the low cost of $6.00 to its subscribers. Why? Because AC believes the Amiga public deserves a publication which, not only provides indepth coverage of Amiga advancements, but also places its readers' interests first.

Amazing Computing for the Commodore Amiga—the choice of serious Amiga users.

Bug Bytes

The Bugs and Upgrades Column

by John Steiner

There are lots of upgrade notices this month, but first here are a few bugs and workarounds to fix some of them.

• • • • • • • • • • • • • • •

Gold Disk's latest release, Professional Page, Version 1.3 provides excellent dot-matrix output, and the Compugraphic fonts display on screen beautifully. As with any major release, we have found a couple of problems and workarounds. Several users found that the location of an object during page layout and the location on the page after printing to dot matrix are not the same. Objects are placed 1/4 inch down and 1/4 inch to the right of where they were placed on the on-screen page. Postscript output is unaffected by the problem.

Gold Disk is aware of the bug and is working on it. In the meantime, there are two possible workarounds. If your printer has a repositionable paper feed mechanism, you can misadjust the top and left margin by 1/4 inch. If you have an HP LaserJet or compatible, an HP DeskJet or another sheet-fed printer that will not let you position the paper, the easiest workaround for the problem is to set the page's left and top margins to 1/4 inch less than their proper position. This will essentially offset the output by the correct amount, and allow the printout to be positioned properly.

There have been a few complaints about the hard-disk install utility in Professional Page 1.3. Some machines had no problems with the installation, yet on others the installation software refused to recognize the ProPage program disk. Users who ran into the problem ended up installing the program manually by copying all the associated files to their correct locations on the hard disk. While I was assisting one person with his upgrade, the reason for the problem came to light. Upon investigation, I found that the install program does not work properly when used with the ARP command set. If you are using ARP to temporarily assign a C: directory that not contain official Commodore-issued C: commands, then the problem is your own and not Gold Disk's fault.

• • • • • • • • • • • • • • •

In EMail this month, I received a letter from Mike Montona. He recently bought a Commodore A590 hard drive. He has an Amiga 500 with 1 Meg RAM and 1.3 Kickstart. When using Excellence! from Micro-Systems Software, the computer would Guru every time he clicked on "Save" or "Save as". It made no difference whether he was running it from the A590 or the floppy disks. He was having problems with several other programs as well. He was able to get in touch with Andy Finkel at Commodore, who had been looking into this problem. According to Mr. Finkel, the programs that were having problems were those that use memory location 0 as a pointer. Excellence! was looking at location 0 and expecting to get a 0 back; any other value at this location would cause a crash. The problem is twofold. First, older versions of the fast file system were setting location 0 as a pointer to GDOS. Second, Mr. Finkel commented that any programs which reference memory location 0 were always in danger of crashing.

The solution to the problem is to update the fast file system to the latest version. Boot from your A590 Install floppy and run HDToolbox. Go to the partition drive page, select Advanced options and select Add/Update File Systems. On the new page, select Update File System and give it a new version number; otherwise leave it alone. Next, click OK on the requester, then OK on the page, and click OK on the partition page. Back on the main page, click on Save Changes to Drive. Then turn your system off, and boot from the hard disk. That should fix the problem. The latest version can be found on the 1.3.2 Workbench distribution.

• • • • • • • • • • • • • • •

I received a letter from S. L. Simons, Jr. of Houston, TX regarding a hardware problem. Mr. Simons installed a Xetec FastTrak SCSI hard-disk system and found that it wouldn't work properly with his Minimegs memory expansion. The technical support representative at Xetec explained that there was a good reason why the two products could not share the expansion bus, and pointed out that Xetec would soon have a memory expansion unit for sale that would work. The M.A.S.T. representative was much more helpful. It seems that the two devices would try to configure at the same time, and the problem was easily solved by installing a diode in the Minimegs according to their instructions. Mr. Simons goes on to comment that the modification is in place on all units currently being shipped.

• • • • • • • • • • • • • • •

Also in this month's mail bag was a letter from George Bailey who commented on problems with the Anim format DeluxePaint III uses. VideoScape 3D anims return a "Not a valid Anim file" requester when DeluxePaint III tries to load them. I have also heard of this problem via messages being transmitted over the electronic networks, and I have run into the problem when loading anims generated by programs other than VideoScape and DPaint III. Mr. Bailey compared the headers of both kinds of Anim files, and noticed that they have different data chunks. If you have found any workarounds to this problem, let me know and I will pass the word around.

Trevor Briggs of Tampa, FL answered the request posed in the September, 1989 "Bug Bytes" regarding Activision. The company was taken over by Mediagenic. He wrote them about upgrades for Music Studio, and after several months has not received a reply. He has also heard rumors of a public domain program that lets you use digitized instruments within Music Studio. If you know of such a program, let me know. I'll pass the information along.

Mr. Briggs also asks if there might be an upgrade to Pagesetter. A quick call to Gold Disk provided the good news. Pagesetter, currently shipping in Version 1.1e, is being upgraded. The spokesperson could not provide any information as to the new features being added or the expected shipping date other than to say, "First quarter, 1990." She commented that details on upgrade policies would be made available when the program is ready to ship.

• • • • • • • • • • • • • • •

Gerry Hull of Creative Focus has released a new version of Better_DJ Version 1.1. The Better_DJ is an enhanced printer driver for HP DeskJet printers. The new release takes care of a problem that causes extra pagefeeds when printing with ProWrite. This version also corrects a problem that could arise with Pen Pal. Better_DJ is freely redistributable, and is found on several information services and bulletin boards around the country. Creative Focus has also incorporated these changes and provides a more fully featured DeskJet driver called Super_DJ.

• • • • • • • • • • • • • • •

Jim Rickman of Interactive MicroSystems sent a notification stating that owners of Media Processor, a video production software package, can upgrade to Version 1.3, Rev 4 at no charge by mailing in their old disks with a stamped, self-addressed envelope.

• • • • • • • • • • • • • • •

Mike Smithwick, author of Galileo, the Amiga astronomy program, wrote to tell us that the software is now being published by Virtual Reality Laboratories. The program has been upgraded to Version 3, and has been renamed "Distant Suns". Improvements to calculation speed, requester design, and general enhancements and bug fixes have been made. Two new databases are also available. Deep-sky objects has over 200 images of galaxies, nebulae, and star clusters. The Skymap database will display 20,000 stars. Upgrades are available to original Galileo owners only through the author. Send your original Galileo master disk and $15.00 (CA residents must include sales tax).

• • • • • • • • • • • • • • •

Digi-View Gold 4.0 is now shipping from NewTek, Inc. The software upgrade to the most popular Amiga digitizer contains an impressive new video mode, 4096-color, high resolution. The new graphics mode is called Dynamic HiRes, and uses a technique whereby the high resolution 16-color palette is changed with each new line. The program also supports image noise reduction, and "fringe-free" HAM modes, ARexx support, super bitmap digitizing, 24-bit color support, and several other new features. The upgrade is $24.95, plus $6.00 shipping and handling. To upgrade, send your name and address and $30.95.

• • • • • • • • • • • • • • •

TV*Text Professional is a high-end video titling program from the Zuma Group. The program supports all screen resolutions except HAM, Amiga color fonts, and color animation effects such as glints, glows, and sheens. The product is available to owners of the original TV*Text as an upgrade in conjunction with Zuma Fonts. If you order your upgrade with Volume 1, 2, or 3 Zuma Fonts, the upgrade cost is $50. The Volume 4 font set and upgrade is $55.00, and an upgrade with Volume 5 fonts is $65.00.

• • • • • • • • • • • • • • •

That's all for this month. If you have any workarounds or bugs to report, or if you know of any upgrades to commercial software, you may notify me by writing to:

John Steiner
c/o Amazing Computing
P.O. Box 869
Fall River, MA 02722

...or leave EMail to Publisher on People Link or 73075,1735 on CompuServe

•AC•

Products Mentioned

Professional Page 1.3
Gold Disk
P.O. Box 789, Streetsville
Mississauga, Ontario,
Canada L5M 2C2
(416) 828-0913
Inquiry # 200

Minimegs
M.A.S.T.
Memory and Storage
Technology, Inc.
3881 Benatar Way
Chico, CA 95928
(916) 342-6278
Inquiry # 201

FastTrak SCSI Hard Disk
Xetec, Inc.
2804 Arnold Rd
Salina, KS 67401
(913) 827-0685
Inquiry # 202

Super_DJ
Creative Focus
P.O. Box 580
Chenango Bridge,
NY 13745-0580
Inquiry # 203

Media Processor
Interactive MicroSystems, Inc.
80 Merrimack Street, Ste. 20
P.O. Box 1446
Haverhill, MA 01831
(508) 372-0400
Inquiry # 204

Galileo Upgrade
Mike Smithwick
25215 La Loma Dr.
Los Altos Hills, CA 94022
Inquiry # 205

Digi-View Gold 4.0 (Upgrade)
NewTek, Inc.
115 West Crane St.
Topeka, KS 66603
(800) 843-8934
Inquiry # 206

TV*Text Professional
Zuma Group
6733 N. Black Canyon Highway
Phoenix, AZ 85015
(602) 246-4238
Inquiry # 245

Roomers

by The Bandito

[The statements and projections presented in "Roomers" are rumors in the purest sense. The bits of information are gathered by a third-party source from whispers inside the industry. At press time, they remain unconfirmed and are printed for entertainment value only. Accordingly, the staff and associates of* Amazing Computing™ *cannot be held responsible for the reports made in this cloumn.]

REFORM

While public attention has been focused on the incredible changes taking place in the Communist Bloc, some equally amazing transformations are happening at Commodore. There are similarities to the current changes in the Soviet Union and at Commodore. After many years of denying that anything is wrong with the economy, the old leaders have been replaced. Many years passed while new leaders were found to pursue the failing policies of the past. Finally, a young, energetic leader with vision has replaced the last of the old hard-liners.

With this new leader has come a series of breathtaking changes in policy. The new leader has moved rapidly to consolidate his power base, tossing out the aging bureaucrats who had proved unable to do the job. With a cadre of new staff, the leader has embarked on a new policy of *Glasnost* (openness) and has proved his mastery of public relations. At the same time, he has begun *Perestroika* (restructuring) to move his enterprise into the forefront of the world economy. Gorbachev and Copperman have much in common, don't they?

Copperman's stated goal is to bring Commodore into the same league as IBM and Apple. To do this, he must enhance Commodore's image, which is one of the main goals of the new ad campaign which uses celebrities such as Tommy Lasorda, Little Richard, Tip O'Neill, as well as astronauts Buzz Aldrin, Scott Carpenter, and Gordon Cooper. Of course, Copperman must improve Commodore's substance as well. This means better distribution, as well as presenting new reasons for people to buy Amigas, and selling more Amigas.

Copperman plans to increase support for software developers in an effort to get more and better software for the Amiga. As part of the incentive plan to get better software, Commodore will step up the pace of product development and improve the speed, resolution, and number of colors offered on the Amiga. Marketing-wise, Commodore will focus on interactive training and desktop video as key markets for the Amiga. Hey, sounds like they've finally noticed where the Amiga really stands out.

"Copperman plans to increase support for software developers in an effort to get more and better software for the Amiga. As part of the incentive plan to get better software, Commodore will step up the pace of product development and improve the speed, resolution, and number of colors offered on the Amiga."

ATTENTION, PLEASE

Meanwhile, the PR blitz continues. From being totally unseen a few months ago, Commodore has gathered a surprising amount of ink in major media outlets recently. Mentions in *Time*, as well as a *Forbes* article stating that the Amiga is tops in multimedia, and a cover story in *Marketing Computers* about the Amiga and Copperman's revamping of Commodore, have expanded the range of Amiga publicity.

In an unusual appearance, Commodore Chairman Irving Gould was the keynote speaker at the Software Publisher's Association meeting in Montreal, Canada. Of course, he hyped the Amiga. The Bandito thought it was interesting that he used "multimedia" as his hook — you know, the same buzzword that Apple's been using in their advertising lately. It's no coincidence that a recent *Business Week* article called Commodore "The Leader in the Multimedia Field". Seems like Apple's advertising is helping Commodore quite a bit by making multimedia a hot topic; Apple can't provide a solution, so people turn to Commodore.

For any of you living in a cave somewhere, the Bandito will now present some information about how Commodore is actually advertising the Amiga. The basis of the strategy is Commodore's attempt to re-position the Amiga as "The Computer for the Creative Mind". Commodore, dissatisfied with previous advertising efforts, retained a new ad agency in May: Messner Vetere Berger Carey Schmetterer (say that three times fast) to develop the television and print campaign.

The print ads display photographs of Amiga users, ranging from a kid making a birthday card all the way to a neurologist. The ads have run in a seven-page clump in *Time*, *Newsweek*, and *Life* magazines, and will appear separately in other magazines. For the most part, reaction among dealers and developers is positive but guarded. Everybody is hoping the ad campaign sells more Amigas, but after many years of

BRIDGEBOARD USERS!

Don't waste money, slots, or desk space buying extra IBM-compatible or Amiga floppy drives! The **Bridge Drive Commander +** gives you direct access to all your internal and external Amiga drives from the Bridgeboard, and direct access to IBM type 360K and 720K drives from AmigaDOS. **Bridge Drive Commander +** is totally transparent and automatic. Put an IBM type disk in any drive and use it just like on any IBM compatible! Put in an Amiga disk and return to Amiga use! Just that simple, just that fast! One drive can use Amiga disks at the same time another is using IBM-compatible disks. Disks are completely usable by other Amiga and IBM-compatible computers. All hardware, no software drivers to load, no precious memory or expansion slots used up. Plugs onto motherboard at internal drive connector. (No soldering or wiring changes.) Compatible with all Bridgeboards (8088, 80286), SideCar, all accelerator boards (any 680x0), hard disks and other hardware and software.

Bridge Drive Commander + ... **$ 97.50**

MJ SYSTEMS

Dept 10A, 1222 Brookwood Road, Madison, WI 53711

1-800-448-4564

(24 hours MasterCard/VISA)

Product names are trademarks of their respective companies.

Circle 149 on Reader Service card.

promises, no one's willing to get very excited. Yet.

By the way, all the partners at Messner, et al. have Amigas — and they actually use them. This has gotta help them create better advertising for the Amiga, since they have some idea of what the machine can do.

Commodore is to continue its advertising blitz into the first quarter, concentrating on A2000 hardware and applications. The objective is to attract the high-end buyer in specialized markets like graphics and video. Developers are stunned by the idea of continuing advertising for the Amiga, since it's something they have not seen before. Now the developers have to figure out how to take advantage of the fact that Commodore is actually trying to sell Amigas.

FACTS AND FIGURES

While Commodore is doing all the right things for 1990, current sales are still in a slump. Commodore announced a first quarter loss of $6.5 million on sales of $165 million. These are down from the $9.6 million earned on $200 million in sales in the quarter a year ago. The press release claimed that this stunning decline in sales was due to waning sales of the Commodore 64. Commodore claimed that sales of both the Amiga and PC clone lines were up. The stock is now floating around 7 1/2. The Bandito remembers when it was closing in on 20 just a few months ago. Looks like the big investors are betting against the Amiga, but the Bandito predicts that the Amiga faithful will be the ones who have the last laugh. However, it looks as though the C64 is finally on its way out, taking its rightful place in computer history. Now if Commodore can just sell half as many Amigas in the next year...

"Commodore is going to continue its advertising blitz into the first quarter, concentrating on A2000 hardware and applications. The objective is to attract the high-end buyer in specialized markets like graphics and video."

While things are going well for Commodore in lining up new distribution, times are tough for publishers of Amiga software. Why? Full-line Amiga distributors (that is, distributors who carry products from all Amiga hardware and software publishers and sell to retailers) have always been rare, and they're getting scarcer. Southern Technologies, for years the biggest distributor of Amiga hardware and software, is reportedly deep in debt and has sold out to avoid complete bankruptcy. Several publishers are owed large amounts of money by Southern Tech, and the sudden uncollectability of that money could put them in a precarious position. The new owners promise some partial repayment, but not more than a fraction of what is owed. This all makes it tougher for a publisher, who now must either deal with individual retailers (a time-consuming task considering their numbers) or become an affiliated label of a large publisher to gain access to their distribution ability. On the other hand, more of the main-line software distributors like Micro-D are starting to carry Amiga products, but only a few are hot sellers. Distribution will likely continue to lag behind Amiga sales for quite a while, even though the market for Amiga products continues to grow.

WHILE WE'RE ON THE SUBJECT...

Does Commodore really intend to go through with the so-called C65? The C128 has already been dropped, and the C64's sales are slowing down so fast it could be down at the level of the Atari ST by Christmas 1990. The Bandito still hears whispers that Commodore intends to replace the aging C64 with the "C65" sporting 256-color graphics and a 4-MHz 6502. The Bandito pleads with anyone at Commodore to shoot this one in the head before it leaves the door. Doesn't anybody remember the Plus/4 fiasco? Look, don't take the Bandito's word for it. Try this simple test: Approach any software developer currently making good money in the marketplace, have them sign the customary non-disclosure agreement so they won't blab your ultra-secret technology, then show them the machine. Now ask them if they will develop software for it. The more polite developers will manage to stifle their laughter before they say NO; but be prepared for most

MICHIGAN SOFTWARE
43345 Grand River
Novi, MI 48050

AUTHORIZED AMIGA DEALER

(313) 348-4477
Call for current prices.

For The A-1000:
•The NEW INSIDER II
•KWIKSTART 1.3 ROM BOARD

For Dealer inquiries, call
(313) 685-2383

Circle 116 on Reader Service card.

developers to just laugh hysterically. Get the message, Commodore? Nobody wants to do 8-bit software anymore! Besides, none of the companies are willing to take any risks on a new software format. Look how hard it is to convince developers to write for the Amiga, a machine with over a million sales and four years longevity behind it. Save yourself a bundle of money and burn all your notes, designs, and prototypes of the C65. Then, put that engineering talent to work on new Amiga projects. Better yet, why not "accidently" leave the C65 plans lying around outside Atari headquarters?

CHANGING THEIR TUNE

In the past, the Bandito has taken a somewhat adversarial tone in talking about Commodore and its actions. This is only natural — Commodore has spent the last 5 years in turmoil, and certainly every Amiga owner, except those who work in West Chester or those who are certifiably deranged (the two sets are not mutually exclusive), has felt that the Amiga has succeeded in spite of, rather than because of, Commodore's efforts. We have seen a procession of presidents parading past us, each promising: Things Will Be Different This Time, and The Amiga Will Really Sell. Yeah, sure. They made some pretty speeches, but the fundamentals didn't change and Commodore went right on being inept.

Well, the Bandito feels that things have really and truly changed. Perestroika is taking place in West Chester. Harry Copperman has conducted a complete purge of revisionist elements and reoriented the Party towards more capitalistic goals like selling more units. There is even hope for the poor dealers out on the collectives; they're seeing some support from the Central Committee. For once, the Bandito supports what Commodore is doing, because they're doing all the right things. More staffing, heavy advertising, more dealer support — it's all happening. Congratulations, Commodore, You're Doing The Right Thing. Now, keep up the good work and don't let the pessimists and cynics steer you off course. It will take time, but the Amiga can take its rightful place as a true competitor in the computer market.



Exciting New Game of Legal Affairs

Courtroom

- Act as Prosecutor or Defense Attorney
- Play against the Computer or Another Person
- Choose from Liberal or Conservative Judges
- Select Criminal Cases from the Court Docket
- Question Witnesses, Raise Objections
- Convince the Jury and Win the Case

only $49.95

from...
FairBrother & SoeparMann
5054 S. 22nd Street
Arlington, Virginia 22206
(703) 820-1954

Circle 179 on Reader Service card.

Now that the Bandito has thoroughly flogged that horse, let's find out what else has been happening. Hmm, it says here that Infinity Software died a quiet death in April, but for the most part, no one cared enough to notice. Gee, at least make a loud noise next time you head out a window, OK? The Bandito hates being behind the times with the news.

AMIEXPO REPORT

According to the Bandito's informers, the big earthquake in San Fransico not only leveled some buildings, it also leveled attendance at the Expo that took place just a few days later. Only a few thousand souls ventured forth to attend the show — some of the exhibitors didn't even show up. But there were a few bright spots. Notably, Elan Designs showed off Invision Plus, a new version of their software for the Live! board that creates some amazing effects in real-time. And NewTek displayed Digi-View 4.0, which offers "dynamic hi-res" with all 4096 colors in hi-res interlace! You could hear the onlookers muttering to themselves about cramming the copper list and similar esoterica. The new software also has an ARexx port for multitasking with Digi-Paint 3 or other software.

In the latest installment of the story, Death of a Publisher, Epyx has filed for Chapter 11. The company is expected to reorganize, pay off debts, and become an AL of some publisher (any bets on Electronic Arts?). Employees are still owed some wages, but unfortunately they come last on the list of creditors. Don't worry for them — all the good ones have already been absorbed by other publishers who are always hungry for talent.

New hot game: ReadySoft's *Space Ace* follows in the footsteps of *Dragon's Lair*. Better compression techniques allow even more scenes. It's still a simple-minded game, but there's nothing better for knocking the footwear off of your friends.

There is a new program called ProVector that is posing a challenge to ProDraw. ProVector has object-oriented drawing tools, is ARexx-compatible and has PostScript output. It's about time for some hot competition over structured graphics — things have been a little slow lately in that field. The Bandito loves a good features war, and it looks like the shooting has died down for a while in the HAM Paint Wars.

(continued)

Memory Management

Amiga Service Specialists

Over three years experience!
Commodore authorized full service center. Low flat rate plus parts.
Proudly affiliated with . . .
The Memory Location
396 Washington Street
Wellesley, MA 02181
(617) 237-6846

Circle 186 on Reader Service card.

KEYCRAFT™ regularly $49.95 now only $29.95

Become a power user with KeyCraft, the keyboard and mouse macro-maker for the Amiga. Customize applications to work the way *you* want them to... Double your productivity... Triple your typing speed... Create realtime animations with any paint program! Act now to receive KeyCraft at this low introductory price (don't forget to ask for your free "PD-Power")

Free Software!

Be one of the first 100 people to purchase KeyCraft or ProPOD from this ad and we'll include "PD-Power", our exclusive collection of power utilities for the Amiga, ***FREE!***

version 1.3 ProPOD™ Programmer's Online Database

regularly $39.95 now only $29.95

C and assembler

Stop searching through those RKMs! ProPOD provides *instant access* to all system structures and function-call templates at the touch of a key! You'll *save valuable programming time* when you let ProPOD "look it up" for you! Guaranteed to increase your coding productivity, or your money back!

Win $2000 in Fish!

We're giving away the **entire Fred Fish Collection** (estimated $2000 value) to one lucky member of the Amiga community. To enter, send a self-addressed, stamped envelope to:

SSI Free Fish Contest
PO Box 59800
Chicago, IL 60659

One entry per person. No purchase necessary. Void where prohibited. Winner's name will be published in the June, 1990 Amazing Computing. Contest ends 4/15/90, and entries must be received by then. Rules will be sent to all contestants (that's why the SASE). **Good Luck!**

StraightLine Software Inc. *The Shortest Distance Between Two Points*
3818 North Kimball Avenue • Chicago, IL 60618
Mail orders to: SSI • Box 59800 • Chicago, IL 60659
When ordering by mail, include $2 s & h. Illinois residents, please add 7% sales tax.

To order, call **1-800-458-2995**
IL & overseas call 1-312-539-5506
Visa, MasterCard, and CODs accepted.
Be sure to ask for your free "PD-Power" disk.
Introductory prices valid until 2/15/90.

Circle 166 on Reader Service card.

The Bandito was checking out the Sears catalog and found that not only do they carry the Amiga, but Sears also sells a few pieces of software — Interceptor, BattleChess, Test Drive II, Nicklaus Golf, Batman, and Double Dragon. Why not WordPerfect, Professional Page, DeluxePaint III, PlanIt!, and SuperBase? Or Digi-View, Performer, TV*TEXT, and Director? What do they think it is, a game machine? Sheesh. At least offer people some choices.

The Bandito has heard of yet another attempt to save the ancient 8-bit technology of the Apple II from extinction. This time, the idea is being floated to "Clarisize" the II — spin it off into its own company, free to swim (or more likely, float like a rock). The problem is that they would be competing with Apple in the under $1000 market, as Apple brings out its low cost Mac next year. A fast IIGS could be sold for only about twice as much — still not competition for an Amiga, but an interesting choice nonetheless. So we may yet see the IIGS Plus. And other sources swear that Apple has a version of HyperCard ready for the IIGS. Supposedly, it's a real productivity enhancer, because there's plenty of time to organize your files and make telephone calls while you're waiting for the software to update the screen.

"Elvis will be found living in West Chester, PA. He will be working as a technician at a local computer manufacturer."— number four on The Bandito's list of predictions for 1990.

DRUM ROLL, PLEASE!

The Bandito's fearless predictions for 1990 are here. Yes, of course, you remember how incredibly accurate last year's predictions were, don't you? Expect the same level of precision this time. Without further ado, here's what the Bandito believes will happen in 1990:

- By the end of the year, the A500 will be selling for $399. A complete system (1 meg, 2 drives, and monitor) will cost less than $800.
- The Video Toaster will finally ship.
- Commodore will be the target of a hostile takeover attempt.
- Elvis will be found living in West Chester, PA. He will be working as a technician at a local computer manufacturer.
- A major software publisher will announce that they are porting their bestselling business software product to the Amiga.
- Commodore will finally announce a CD-ROM drive for the Amiga.
- The A500 will come with 1 megabyte of memory standard by the end of the year.
- Amiga sales in the U.S. will surpass Amiga sales in Europe.
- The A3000 will ship in the Spring, though people will complain that the price is too high. Commodore will let slip that they are working on a 68040-based version.
- Commodore will strike a deal to bring a standard font-description language to the Amiga.
- The new 640 x 480 non-interlaced resolution mode will create more interest in the Amiga for word processing. New versions of major word processors will take advantage of this, becoming competitive with the best programs on other machines.

The Bandito leaves a review of last year's predictions as an exercise for the reader. Before you laugh too hard at the Bandito's accuracy, check out what psychics were predicting for the year. Makes the Bandito look pretty smart, eh?

•AC•

Facing the CLI.

Disk Structures and Startup-Sequences

by Mike Morrison

One of the problems people who use the Workbench (WB) have while trying to learn the CLI is the confusing difference in terminology. If you are a WB user, you might say something like "DPaint is in the Graphics drawer", while a CLIer would say "DPaint is in the Graphics directory". These two statements mean the same thing. This example may be somewhat subtle, but this is just the type of thing that causes confusion.

WORKING WITH PICTURES

When using WB, everything is icon-based. This means that you are working with each entity by manipulating its associated picture.

Example: You have a diskette that you want to copy.

1. Put the disk you want to copy into drive 0.
2. A disk icon with the name of the disk will appear on the WB.
3. Put the blank disk into drive 1.
4. A disk icon with the name of the disk will appear on the WB (it may be 'DF1:BAD' if you are using a new non-initialized disk).
5. Drag the icon of the disk you want to copy onto the icon of the disk in the other drive.
6. The system then copies the disk in drive 0 to the disk in drive 1.

When using the CLI, you must tell the computer what to do by using commands. To accomplish the same thing above using the CLI you would use the command:

1> diskcopy df0: to df1: <return>

As you can see, when using the CLI you must talk to the computer in words. First, you have to tell the Amiga what you want do — in this case copy a disk (diskcopy), and you must tell it exactly what to copy (df0: to df1:). This is exactly the same thing we did when we dragged one icon over the other on the Workbench. If you aren't used to the CLI, the Workbench will seem much easier. It is important to note, however, that there are certain things you just cannot do from the Workbench, and there are other times when you may have the option but using the CLI may actually be faster, easier, or both. In any case, if you are ever going to use the CLI, understanding the disk file structure will make using it much easier.

Just like a filing cabinet, a disk has drawers within it...In fact, you can open a drawer and the only thing inside of it might be five other drawers!

CLI & WB BLUEPRINTS

The WB uses icons to represent disks, drawers, files, etc. These are related to one another in a certain manner. A disk would be the topmost level, whether it is a floppy disk, hard disk, or optical drive. Let's use the a filing cabinet as an analogy. The disk would be the filing cabinet that holds all the data. Just like a filing cabinet, a disk has drawers within it. Inside a filing cabinet drawer, there are file folders. Inside a drawer on a disk there are also files.

The analogy between a disk and a filing cabinet is not perfect. When you open a disk (by double-clicking the disk icon), you may see both drawers and files (programs, letters from your word processor, data from your spreadsheet, etc). This is the equivalent of storing papers loosely in the filing cabinet instead of in one of its neat little file folders. Also, on a disk, you can have drawers inside of drawers, inside of drawers, etc. In fact, you can open a drawer and the only thing inside of it might be five other drawers!

This analogy also works when using the CLI, only there are a few names that have been changed (to protect the innocent, I suppose). Instead of drawers, the CLI lists directories. Take a look at Figure One. This really is how the disks and their directories and files are laid out. This is really how all Amiga disks are structured. Workbench sometimes makes it easier to get around because it has visual aids (icons). This makes understanding the underlying structure less important. When using the CLI, understanding this filing structure is vital, but once you get the hang of it, it will be a piece of cake!

HIDE 'N' SEEK

When you boot up your Amiga, the disk that you booted it with becomes the System disk. This is where Ami will look for commands, fonts, libraries, devices, and more. In the last article we talked about how to get around the disk-swapping problem when using commands from the

CLI. We put certain commands into a RAM disk and told the system how to find them. As promised, I'll tell you how to add these commands to your startup file.

STARTUP-SEQUENCE

The startup-sequence is one of the files that is searched for when your Amiga boots up. It is located on your WB disk in a directory called S. The S directory is where startup scripts are kept. A startup script is a file which contains a list of commands that are executed when a particular program is started. Usually this list (when executed) will set things up in a certain manner. The startup-sequence is the startup script the Amiga executes upon boot-up to arrange things the way you want. Most people don't even know that it exists and use the default script (the one Commodore included with your Amiga). This script is O.K., but why not customize it?

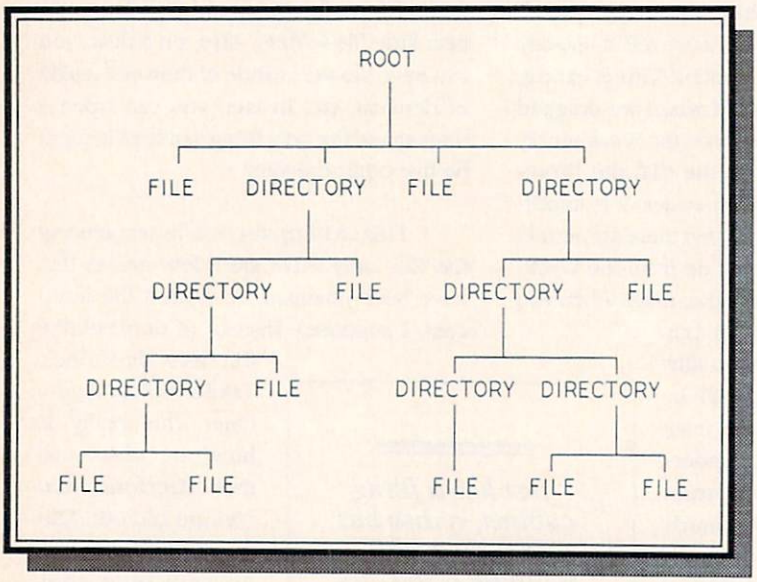


Figure One: A sample disk structure showing a possible relationship between directories and files.

CUSTOMIZING YOUR STARTUP-SEQUENCE

The usual rules apply: Work with a back-up of your already backed-up WB disk. *Never work with the original.* Playing with the startup-sequence can cause your Amiga to not want to boot up to Workbench. If you use a copy and make a mistake, you can always revert back to your original copy of WB.

Get into the CLI by double-clicking in its icon (in the Utilities drawer on your WB disk). Once you have the CLI prompt, type:

ed s/startup-sequence <return>

This starts the editor called 'ed'. The computer finds ed because it is in your C directory. The 's/startup-sequence' tells ed to open the file called startup-sequence that can be found in the directory s. Note: if ed comes up blank with a message that says "opening new file", you probably made a typo while typing "startup-sequence". Ed is smart because if it can't find the file you told it to find, it assumes you want to create one. We don't want to create a new file, so if ed answered "opening new file", hit the 'esc' key (upper left) and then the 'q' key. Ed will go away and you will be back at the CLI. Try the original command again.

The startup-sequence is now in ed. Ed has a whole list of commands which you can find in the manual that came with your computer. Use the arrow keys to move around in the text. This is the script that the Amiga executes during boot-up. Many things happen here. Many of these are AmigaDOS commands; some are actual programs. The startup-sequence executes in order from beginning to end (there are commands for skipping lines). Scroll all the way to the bottom with the arrow keys (using the Ctrl key and the U or D keys will scroll the display up or down one screen at a time).

Notice this last command:

endcli > NIL:

You should recognize the 'endcli' command from the last article. The ">NIL:" part of the 'endcli' command is a little difficult to explain. Basically, it says "send any garbage floating around into the bit bucket". The ">" causes all output to be redirected. Here the output is redirected to the NIL: device. NIL: is a device similar to DF0:, except that if you say 'copy stuff to DF0:' the file named "stuff" will be copied to the disk in DF0:, and if you say 'copy stuff to NIL:' the file named "stuff" would be copied to nothing, the bit bucket, file heaven, etc..

To add commands to your startup-sequence, add them before the endcli command. Remember to use a copy of your boot disk in case we mangle the startup-sequence (chances are if you make a mistake, Ami won't boot up to Workbench). Use the listing from last month to create a RAM disk with commands on it. When you are done typing in the changes, press the 'esc' key and then the 'x' and <return>. This saves the newly modified startup-sequence to disk. Wait until the disk light goes out and then reboot your Amiga to see if the new startup-sequence works.

To customize your startup-sequence for the world to see, use the 'echo' command. Add something like this:

```
echo "This is Mike's Amiga booting."
echo " "
echo "Have a great day!"
```

To have even more fun, you could change the echo command to the say command and each time you boot up, these lines would be spoken. Usually after hearing a sentence 5 or 6 times you find yourself changing the startup-sequence back to echos. Both echos and say will slow down your startup-sequence, so use them wisely.

Often, using the CLI can be a trying experience, but with patience and persistence you will soon be getting around with little difficulty. Sometimes looking at other people's startup-sequences or script files can help. Enjoy.

•AC•

Dear Animators

January 1, 1990

Dear Animators,

For all of us animators (professional and otherwise) and aspiring animators, 1989 was a very good year. It was the high point (so far) in the history of Amiga animation. Even as we are trying to get this issue to press things are still hopping. We have just received beta copies of Deluxe Video III (which should be officially available soon after you read this), as well as upgrade 2.09 of Sculpt-Animate 4D and we also received word on a brand new animation package to be available in February (keep reading)!

Animation was recognized very early on as one of the Amiga's strongest features and greatest potentials. Remember, even high-end 3D graphics machines like the CubiComp (around $50,000) can't do real-time animation! From the originals like Aegis' Animator and the early animation products of Martin Hash to this year's rash of Animation upgrades and introductions: DeluxePaint III, PhotonPaint 2.0, Sculpt4D, TurboSilver 3.0, MovieSetter and more. The market for animation products is continuously growing and the products it has encouraged are many and varied. Just about everyone has finally gotten into the animation act. The shocking truth of the matter is that, for the most part, they all have something useful to offer.

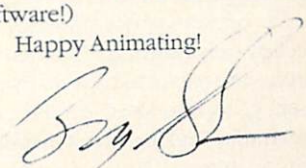
Martin Hash now has an entire line of excellent animation programs and utilities. Most of the popular 3D programs include animation facilities: Aegis' Videoscape, Byte by Byte's Sculpt series, Impulse's Silver series, PageRender 3D from Mindware, Forms in Flight from MicroMagic, 3D Professional from Progressive Peripherals and the Caligari series from Octree. Two of the most popular paint programs, DeluxePaint III and PhotonPaint 2.0 include animation. Zoetrope from Antic is also a combination animation/paint program. We even have MovieSetter from Gold Disk which can have an amateur producing cartoon-like animations, from the included backgrounds and animated clip-art, in a matter of minutes!

A number of programs allow combination of individual frames into animation: Elan's Performer, Video Effects 3D by InnoVision, Animation Station by Progressive Peripherals, Pageflipper Plus F/X by Mindware, The Director from The Right Answers Group and Aegis' Lights, Camera, Action!. MicroIllusions has an entire Line of PhotonVideo products to complement its PhotonPaint 2.0. We even have animated fonts from Anivision as well as Build 'em-clip-art designed to be animated from the PictureBox.

In the last few years even hardware has joined the parade. There are now a number of frame buffers available (and several more coming out early in the year) which offer 24-bit output (professional video quality) to videotape for the many programs that now support this. There are also frame controllers which interface with professional film and video recorders for single-frame animation.

The really good thing about this is that as more excellent animation programs become available for the Amiga the interest in Amiga animation will grow, and as the interest grows it will continue to spur the development of new and better products! Personally, I can't wait to see what 1990 will bring. (In case you can't wait either I'll give you a hint of the future to come. Watch out for the February release of THE ANIMATION STUDIO from Walt Disney Software!)

Happy Animating!



Barry Solomon
Video Editor

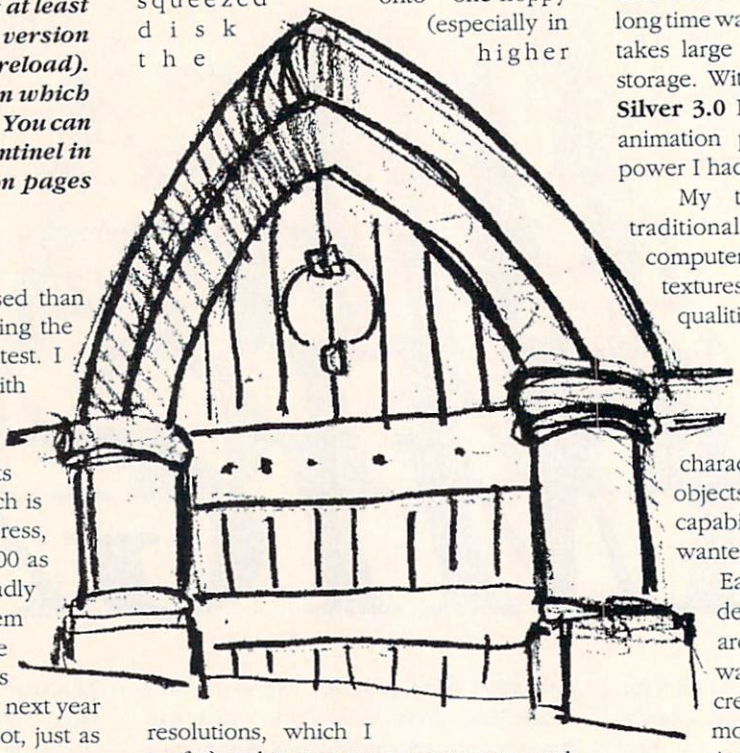
Ladies And Gentlemen, we give you...

THE WINNER!

AMIGA

by Bradley W. Schenck

Bradley W. Schenck won the 1988 BADGE Killer Demo Contest and has just won the 1989 Contest with his new animation, The Sentinel. The Sentinel is available on Fred Fish Disks #263 and #264 (yes, it takes most of both disks). The file is so large that it takes at least three megs to run (and this version has to pause in the middle to reload). There is also a 3.5 meg version which will play without interruption. You can see a few still shots of The Sentinel in our PD Serendipity column on pages 56-57. — Ed.



No one was more surprised than me, when I found myself entering the 1989 BADGE Killer Demo Contest. I had won first place in 1988 with *Charon*, and had decided afterwards that it would be best to leave the contest to other artists the next year. The contest, which is well publicized in the Amiga press, traditionally offers an Amiga 2000 as the first place prize. I had badly wanted to upgrade my 500 system when I entered the contest the first time, and when I won, it was a break for me. I figured that the next year should to be someone else's shot, just as 1988 had been mine.

The rules for the 1988 contest were pretty specific. Entries were required to run on a one-megabyte system and could occupy only one disk; this was because the demos are meant to be used by dealers and users groups to show off the capability of the Amiga, and it was felt that having them run on normal systems with a stock Workbench disk in the other floppy drive would allow these demonstrations to be run on practically any system. When I created *Charon*, many tough decisions were presented by the environment I had to develop the demo for; there are limits to the amount of data that can be squeezed onto one floppy disk (especially in the higher resolutions, which I prefer) and it gave me a structure to work within. Compromises had to be made.

When the rules for the 1989 contest were released, I was amazed to find how easygoing they were. There were no limits on the size of the animations or the memory that would be needed to run them. I had a sinking feeling that we would see a whole slew of demos along the lines of the *Walker* and *Walker II* demos by Brian Williams, which basically would make the previous year's entries look pale by comparison. I started to get this itch to enter after all.

As soon as I had received my Amiga 2000 system, I had given it another two megabytes of memory and a hard drive. One of the things I had wanted to do for a long time was to work in ray tracing, which takes large amounts of RAM, time, and storage. With the introduction of **Turbo Silver 3.0** I had found a rendering and animation program with much of the power I had been looking for.

My two dimensional work in traditional media and, later, with computers, has always depended on textures and surfaces with many qualities. Silver allows me to create a wide variety of surface types, to map IFF images onto objects, and to use prefigured textures that lend certain characteristics and patterns to my objects. I needed that type of capability to create the 3D world I wanted to make.

Early in the spring, I started designing a hallway with an arched door and columns. This was one of the first scenes I created with Silver. For many months, I worked on it when I had time. Soon I had borrowed a Midget Racer from a friend to speed up my rendering times. Using the floating point version of Silver, I set to work on an animation I hoped to enter in the BADGE contest. The early versions of the opening scene were coming along very well, and finally I decided to test the ANIM files on a stock Amiga system. I trotted into my wife's studio to run the animations on my old 500

system and got a nasty shock. The ANIM files had played back beautifully on my system, but were slow and jerky on a stock Amiga.

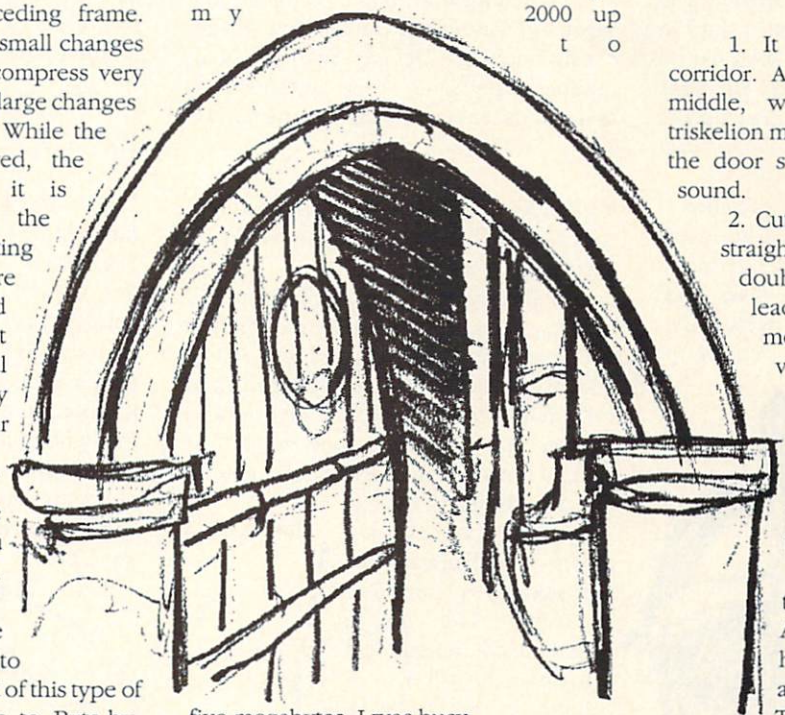
Although the Midget Racer isn't the fastest accelerator on the market, there are a few things it can do swiftly even though it doesn't use 32 bit RAM. Obviously it used a 68881 for floating point math, which was the reason I was using it, but it also has a dramatic effect on the speed of the compression and decompression of data.

ANIM files rely on compressed data. They are a type of storage called delta compression, in which only the first frame of an animation is stored completely. Subsequent frames are stored only as differences from the preceding frame. Frames which contain very small changes from one to another will compress very well, while frames that have large changes will not compress as well. While the ANIM file is being played, the program that displays it is constantly decompressing the frame changes and constructing the next frame. That's where my own system excelled and where the stock system just couldn't keep up. The awful truth is, you can get away with bloody murder in your ANIM files if you have an accelerated system. But if your animation needs to be played on stock systems, you can't take advantage of that.

So there I was. There are a few things you can do to optimise the playback speed of this type of animation file (this applies to Byte-by-Byte's Movie file format as well). The basic idea is to change fewer pixels in each frame. You can do this by lowering the resolution of your screens (that will automatically halve the the amount of data) or by tailoring the animation to make smaller changes from one frame to another. The first alternative is always a last resort for me. I want my work to use the higher resolutions whenever possible, for the smoothest obtainable appearance. The second alternative seemed more sensible. I was animating a large moving object over a reflective floor with dramatic lighting that cast huge shadows. This made for large changes from frame to frame.

Over the next week or two, I experimented with the scene in low resolution, with a non-reflective floor, and with different lighting. I found that the compromises were killing the image, and as it turned out, I went back to concentrating on commercial work and just let the idea sit. It looked as though I wouldn't be entering the contest after all.

The contest deadline was set for September. I waited with some anticipation to see what the other artists had come up with. In the meantime, I had finished a set of tutorials for using the Textures in Turbo Silver, which Impulse had agreed to publish, and I was spending most of my 3D time working up a library of objects to make my world with. I wanted a large number of building blocks for my animations, many little objects to use when the time came. I brought my 2000 up to



five megabytes. I was busy.

I kept finding myself showing Turbo Silver to other users. It's a difficult program to master and, although I hadn't done a huge amount of work with it, many people I knew felt that I had gone far enough with it that I might save them from some of the mistakes I had made. As far as experts go, my money's on Louis Markoya; but he's on the other side of a pretty large continent, so they had to make do with me. I began to get itchy again. I finished one late night session with Leo Schwab and Rick Unland of Commodore, and the next day I heard that the BADGE contest had been postponed until late October, and that there were at that time fewer entries than prizes. Although there are always a batch of last-minute entries, this sounded awful, and I had just been feeling like getting back to work....

I had about four weeks — not much time at all. The story for The Sentinel came to me pretty much immediately; it was built on some of the objects I had been working on, and the setting for the first scene was the same hallway I had started with months before. There would be about five scenes altogether, with certain elements being used more than once. As I saw it I had just enough time, if nothing serious went wrong.

Here's how the animation goes. I'm hoping that you've already seen it, since I hate to hear the plot of a movie before I get a chance to see the film, but I want to explain how the scenes go together.

1. It opens with a view of the corridor. An arched doorway is in the middle, with a crystal pyramid and triskelion medallion at right. After a pause, the door swings open with a creaking sound.

2. Cut to a view past the door and straight down the hallway beyond. A double row of columns and urns leads into the darkness. After a moment, something becomes visible at the end of the hall: a columned brick wall behind a pedestal. A shield hangs on the wall; a helmet is on the pedestal. The atmosphere throughout has the feeling of the Bronze or early Iron Ages.

3. Cut to a closeup of the wall, shield and helmet. After an establishing pause, the helmet rises from the pedestal and turns to face the shield. There is a flash of lightning and the roar of thunder; when the image clears, the helmet is facing us and the shield has moved in front and to the side.

4. The wall is gone, replaced by blackness. It's as though a ghostly figure has risen to face us. At the other side, a sword appears. The helmet swivels to face it, and the sword is snatched up. The helmet turns back to look at us, and the sword sweeps high overhead.

5. Cut back to the hall where we began. There is a crash; the door slams shut, its knocker banging ominously with the impact.

Altogether there are about sixty frames of animation involved, and the entire project was finished within three and a half weeks. How did I do it? I cheated like mad. I always do.

Making the Most of a Little Time

I was able to take advantage of some features of my software to speed up the rendering process. Turbo Silver allows the user to Set a Zone to be rendered: instead of creating a full screen image, one can render only a little 'window' on the screen. This can save many hours of ray tracing.

You have to be careful to take shadows and reflections into account. During an animation perhaps only a small object will be moving, but its shadow will also move, and its reflections (if any) in nearby objects will change as well. Turbo Silver's Solid Model mode provides excellent shading, but without ray tracing at least one frame of your animation, it may not be possible to foresee what the shadows and reflections will be doing.

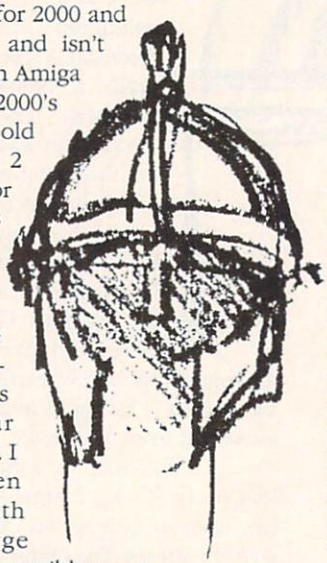
When setting the Zone for a series of frames, you must be sure that you include all the area that will be affected. In the software's current version, you can't set the zone beyond the non-overscan area of the screen (if using overscan), so in some cases you'll just have to render the entire picture.

When it came time to put the pieces together I was able to use a couple of very handy features in Deluxe Photolab's Paint module. The Save From and Load At commands let you save a rectangle of the screen as a brush or as a small screen, and then load it in at another precise screen location. This is especially useful in animations.

I was able to Save from my rendered zones, and to use Load At to load them automatically into position on the full screen frame I used for a background. This was much more accurate and simple than picking up the zones as brushes, clicking each in place by hand. In this way I saved days of rendering time at the cost of a few minutes' work.

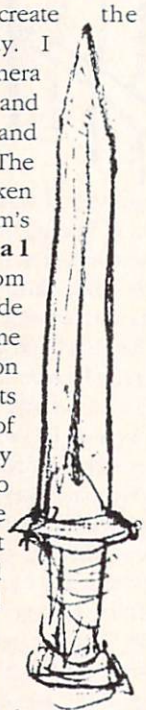
B.W.S.

Although the rules hadn't specified a rigid structure for animations, I imposed some restrictions myself. The demo ought to be contained on two floppy disks, and should run in three megabytes of memory. Three megs is a fairly common configuration for 2000 and 2500 systems and isn't too unlikely on Amiga 500's. The 2000's often use the old Commodore 2 meg board, or the newer 8 meg board populated with 2 megs, and there are several third-party boards with similar configurations. I had been working with several large ANIM files and possibly some stills, and I wanted all of the data to be in memory at the same time, so there wouldn't be any delays due to disk access. The animation would be played with The Director. The Director allows me to use practically every trick that's available on the Amiga, and I have yet to find another program that allows me anything like the control it does over every aspect of a presentation. I have even written a couple of utilities with it. There would be some sound effects, and the animation would not need more than 512K of chip RAM. That is, it wouldn't require the fatter Agnus chip. And it had to be finished in four weeks, or I would miss the contest.

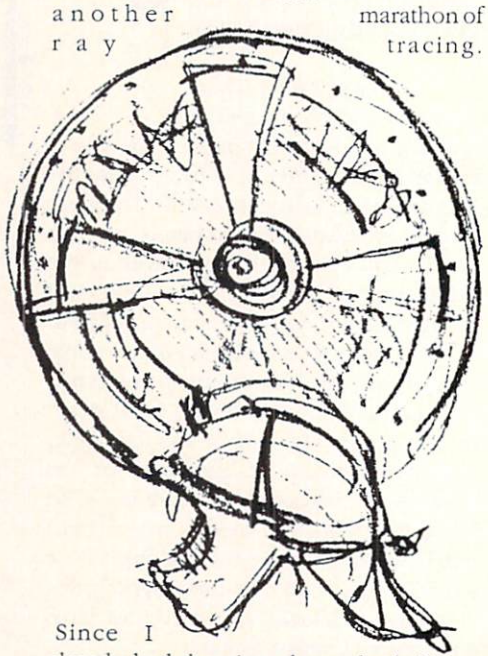


I already had the first scene set up, I only had to create and render the cell files for the animation. I was working in 352 by 440 pixel resolution and overscan, and each frame took a number of hours to render. The first scene took about five straight days to generate; I wasn't able to use any shortcuts for that one.

Next, I had to create the view down the hallway. I repositioned my camera in Turbo Silver, and added the columns and urns down the hall. The columns were taken from Steve Schram's **Architectural Objects** disk from Impulse; I had made the urns some time before with 3Demon (Mimetics), which lets you paint the faces of an object individually - it's nice to be able to see what you're doing! The scene at the end of the hall came next. I had made the wall and shield for a still image, several months before; the helmet was the object I had been working on when I started the animation. There is an IFF image mapped onto the shield, and the brick wall uses Silver's Brick texture. Parts of the helmet had been created with Access Technology's **Digiworks 3D**, which translates IFF images into extruded 3D cutouts. I mapped an IFF image onto the pedestal and placed rings on the wall which were near duplicates of the door knocker from the first scene. After adjusting the lighting for awhile, the stage was set.



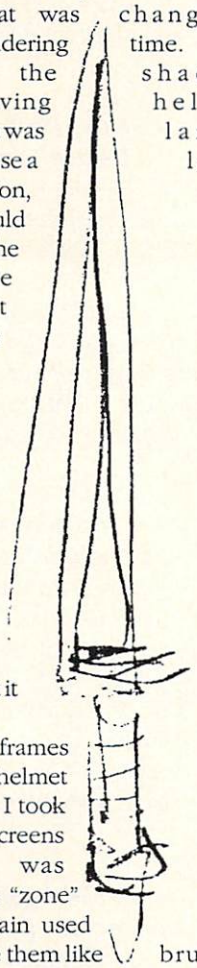
I rendered the first frame, where the helmet is sitting on the pedestal, saved it and copied it to a floppy. Next I started another marathon of ray tracing.



Since I already had the view down the hall and the wall scene done, I was able to take them over to the my wife's 500 system and create the second hallway frame. I used Deluxe Photolab to match the Shield

picture to the palette of the hallway picture and then shrunk the shield picture down to fit at the end of the hallway. I pasted it into place and finished the graphics for the second scene while the third one was being rendered.

I was able to start cheating now. Turbo Silver lets you Set a Zone of the complete screen, and it will render only that zone if you tell it to. I was able to use this feature to render only the area of each frame that was changing, saving some rendering time. One problem was the shadow that the moving helmet cast on the wall; it was large, which made me use a large zone. In addition, the large zone would end up making the final ANIM file larger, and it would also make the file play back more slowly, since it now made greater changes from frame to frame. I could live with that because I already knew what I was going to do about it later.



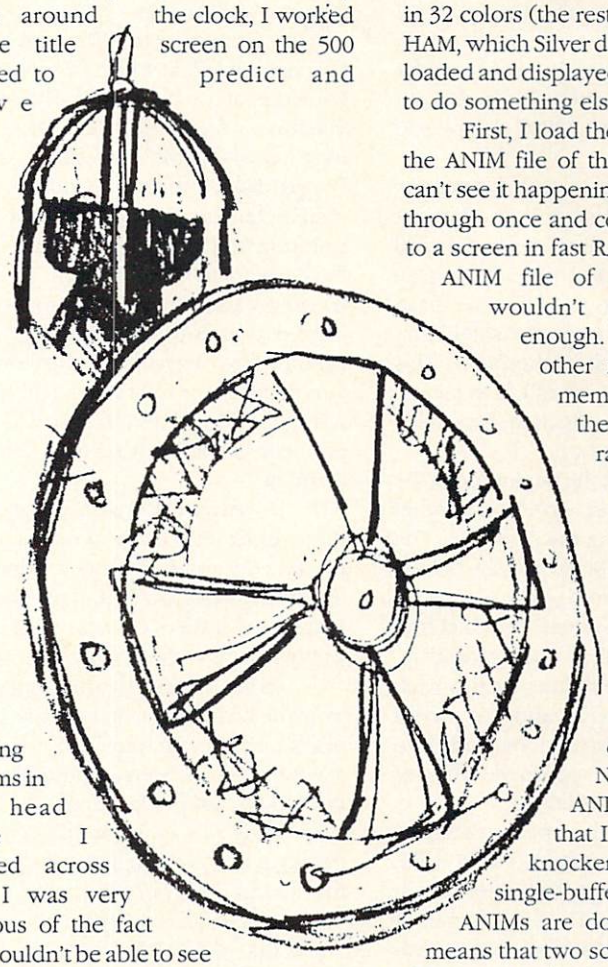
Once the frames of the rising helmet were completed, I took those partial screens (remember, I was rendering only a "zone" of each) and again used Photolab to paste them like brushes onto the first frame (like a background). With a little touching up I had the third scene done. It had only taken a few days.

I had to get rid of the wall: I already knew that. I mentioned that the objects' shadows were causing me a couple of different problems that would affect both storage space and speed. I had decided on using a thunderclap and lightning flash to make a dramatic transition from the third scene to the fourth, and it gave me a chance to lose the background in a way that made some kind of sense. A dead black background would show no shadows and make my playing speed faster. Also, it would enable me to use the Set a Zone feature more effectively.

The lightning flash uses three screens. One shows the helmet and shield in their new positions against the wall; another shows them positioned in the same way, with a black background; and the last one is a pure white screen that creates the flash. I only needed to render the first two — the flash screen was easily made later, with DPaint.

That done, I designed the animation for the sword sequence in wireframe mode. This was a critical step, and I was beginning to feel the deadline pressure already. I adjusted it and reviewed it several times, until I felt it would do. The animation could have used many more hours of design time, but I didn't have them to spare. I started the last long rendering marathon.

While my system was ray tracing around the clock, I worked on the title screen on the 500 and tried to predict and solve upcoming problems in my head before I stumbled across them. I was very conscious of the fact that I wouldn't be able to see the animation on a stock system at home (the 500 system has only one megabyte), and the animation's playback speed on a stock system could still be a problem. I had to know just what I was going to do with my data once it was all ready.



With the fourth scene rendered, I was nearly done. I had already decided to reuse some of the frames of the door opening when the time came to slam it shut; and I was able to set the rendering zone to create just the area with the knocker and its shadow for the frames of the knocker banging against the door. There were only three of those frames, and they were ready in a few hours. I still had over a week to go.

All of the animation files correctly went together pretty much the first time. I had a couple of minor problems, but a silent version was running by the time I crawled into bed that night (or was it morning?).

I had had to plan the animation ahead of time (and I had time on my hands, while it was rendering), and I made some decisions while making it work. Here's a short description of how The Sentinel works:

The title screen is a single IFF image, in 32 colors (the rest of the animation is in HAM, which Silver does beautifully). It isn't loaded and displayed at first, because I had to do something else before I loaded it.

First, I load the sound files and then the ANIM file of the opening door. You can't see it happening, but I play the ANIM through once and copy every other frame to a screen in fast RAM; this is because an ANIM file of the slamming door wouldn't play back quickly enough. By saving out every other frame this way, into memory, I was able to flip the pages much more rapidly than I could play the ANIM file, so the door would really look as though it were slamming shut. This approach used much more memory than an ANIM file, but it was the only way I could get the door to slam at speed. Next, I load in a smaller ANIM of the three frames that I use to bang the door knocker at the end. This was a single-buffered ANIM. Almost all ANIMs are double-buffered, which means that two screens are used to play them: one is displayed while the other is updated to make the next frame. Single buffered ANIMs can only be created with the commercial version of MakeANIM, available on the Director optimize disk. They are meant to store data that you won't be playing normally, but which you want to compress to save space. I then decompress this ANIM just as I did with the first, so that those three frames are also available in fast RAM.

Now the title screen is loaded and displayed, fading in from black. I waited until after I was done with the ANIM files to avoid fragmenting memory into small chunks. The title screen remains displayed until all the rest of the data has been loaded.

The ANIM files for the hallway scene, rising helmet, and the sword swing are now loaded, with some still frames for the lightning flash. Once everything is in memory, the animation starts.

The first frame of the door animation is displayed, showing the door closed. There's a pause so the viewer can have a look at the scene. Then the door animation is played through, with a sound effect for the creaking door. There is another slight pause, and we cut to frame one of the hallway ANIM. After another brief pause, I dissolve in the scene at the end of the hallway. We then cut to frame one of the rising helmet ANIM and hold there for a moment. Then that ANIM is played through.

The thunderclap is made by playing the same 20K sound sample at different speeds, through all four channels, and fading it out as a distant rumble. During the thunderclap, I flip between the three flash pictures, finally settling on the shield and helmet against a black background. This was easily the "cheapest" effect to create, but possibly the most effective. Especially if the volume is turned up.

This places us at the beginning of the next ANIM file.... almost. First, I use another dissolve effect to bring the sword in. The first frames of the ANIM play, as the helmet turns to face the sword, and we pause again; then the next frames, in which the shield moves out as the sword is picked up. Pause again, then the helmet rotates back to face us; pause again, and the sword swings up. All of that is done with one ANIM. The timing is created in the Director script in my playback routines.

Now we cut to the opening scene. I rapidly move each frame into chip RAM, display it, copy the next to another screen in chip RAM and so on. The three frames of the banging knocker are played in several sequences to make the knocker hit the door three times, synchronized to another sampled sound. After a moment, we return to the title screen and start again.

I was still about a week early, and had the animation done. Sort of. I found, that it didn't fit in three megabytes after all. Even without loading the Workbench, there just wasn't enough memory to handle it. The biggest single problem was all of those screens used for slamming the door, not as an ANIM file but as separate images in memory.

There were several ways around it, but none that were attractive. What I finally did was to watch the demo and read the script over and over until I had pinpointed a few important points where I might be able to use a load from disk. These had to be moments where there could be a slight pause, but they also had to fall at points where I could drop some data from memory (not needing it until the demo repeated) and load in something else that I would only need temporarily. This proved to be the best solution.

The final result was two different Director films, one for three megabyte systems and one for systems with more than three megabytes. The latter was my original version, in which all of the data was loaded in at the beginning. The three meg version worked almost the same way, as long as the demo was being run from floppy disks; running it from a hard disk means that some pauses will be shorter, unfortunately. To get around that I would have had to either provide another version based on an average hard drive speed, or ask the user at the start if the demo were running from a hard drive. However, work was picking up, and I needed to give it my full attention. So I decided to call it finished, and got it off in the mail before the deadline.

In retrospect I wish that I had had more time, especially when it came to plotting out some of the animation. To tell the truth, I was amazed and pleased that I'd gotten it finished on schedule, and I'm pretty happy with the way it turned out.

So what next? I figure I ought to retire from the BADGE circuit. I see it as someone else's turn now, another artist – especially a young one, who has more ability than cash flow, and who might be able to surprise us all with something wonderful, getting a new system in the bargain. Does that sound like you? I hope so.

For me, I have another itch now to do work that doesn't depend on available RAM, storage, or playback speed. Going to video looks pretty attractive. Going single frame to video looks better. And with a couple of frame buffers on the market that will let me use two million colors on a screen, it looks better yet. In fact, it all looks pretty good.

•AC•

Products mentioned:

Turbo Silver 3.0
Impulse, Inc.
6870 Shingle Creek Parkway
Minneapolis, Minnesota 55430
(612) 566-0221
Price: $199.95
Inquiry # 222

Deluxe PhotoLab
DeluxePaint III
Electronic Arts
1820 Gateway Drive
San Mateo, California 94404-2497
(800) 245-4525,
in CA (415) 571-7171
Price (Deluxe PhotoLab): $149.95
Price (DeluxePaint III): $149.95
Inquiry # (PhotoLab) 223
Inquiry # (DPaintIII) 224

The Director
Director Toolkit
The Right Answers Group
P.O. Box 3699
Torrance, California 90510
(213) 325-1311
Price (The Director): $69.95
Price (Toolkit: $39.95
Inquiry # (Director) 225
Inquiry # (Toolkit) 226

CYBER Graphics
Architectural Design Disk
Antic Software
544 Second St.
San Francisco, CA 94107
(415) 957-0886
Price: $34.95
Inquiry #244

DigiWorks 3D
Access Technologies, Inc.
P.O. Box 202197
Austin, TX 78720
(512) 343-9564
Price: $129.95
Inquiry # 243

Animation for Everyone

AMIGA

or: Animation Can Be A *Moving* Experience!

Tutorial by Barry Solomon

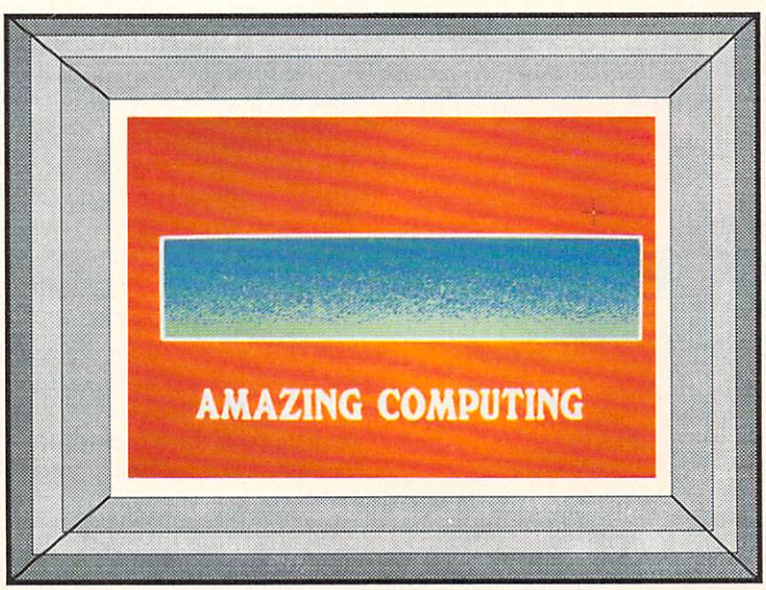
Well, the Holidays are over and I, for one, am so full of turkey and ham that I can barely move. Luckily, however, my finger movements are not impaired so I can share these bits of information with you. As you know, if you read last month's Video Schmideo (and who didn't?!), DeluxePaint III was my pick for best overall Video/Graphics product of the year. DeluxePaint III has some features that can only be described as absolutely unique. The brilliant Move requester is one of them. It has finally brought animation to the masses. So, for those of you who got DPaint III from Santa or have just purchased DPaint III with some of that holiday cash(!), let's take a little walk through the heart of DeluxePaint III's animation feature. Please remember, this is not the advanced class, so if I bore you please feel free to go straight to the next class.

Ready? Okay. Boot up your machines with DeluxePaint III. If you do not already have an extra formatted disk to capture the incredible animation you are about to produce, please format one now. Before we get started, turn on the Coordinates Feature (more about this later). When presented with the Choose Screen Format requester pick Hi-Res (640 X 400), Overscan, and four colors (two bitplanes). Make sure your click on Overscan (which will set the screen size to 704 X 480) and four colors as neither are default settings (16 colors and no overscan is the Hi-Res default setting).

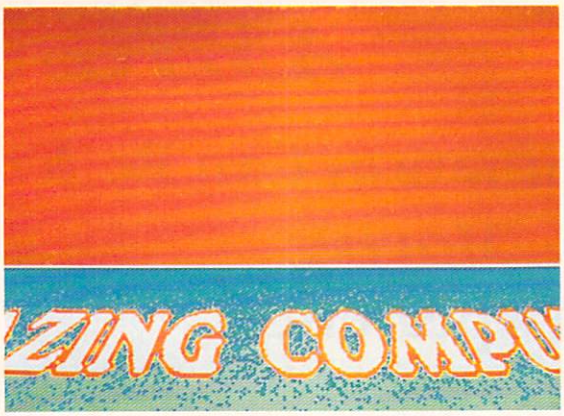
Once your screen opens, hit 'p' to bring up the palette requester. The colors we are going to set can be changed even after the animation is completed, but for now, set them as I did so you know to which colors I am referring. Set the first color (top left—color 0) to red(Red—15, Green—0, Blue—0). Set the second color (bottom left-color 1) to white (Red—15, Green—15, Blue—15). Set color two (top right) to a blue (R—0, G—5, B—15) and the last color (bottom left-color 3) to a green color, R—6, G—15, B—3.

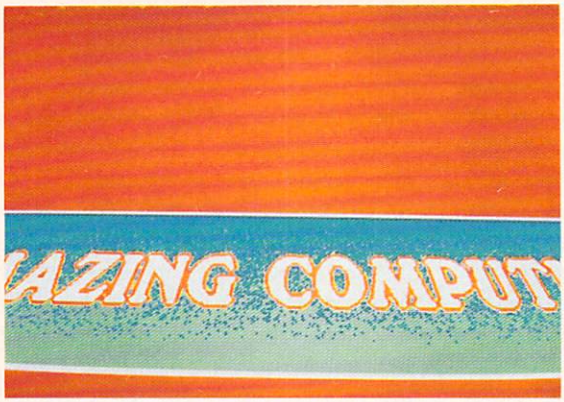
If you are absolutely brand new to the Amiga this may be a little confusing. Just remember that colors are numbered from zero on, so if you have four colors, they will be number zero through number three. Also remember that no matter how many colors you are using in any DPaint resolution, they are numbered from left (top to bottom) to right on your palette. Thus in a Lo-Res, 32-color screen, ..there will be four rows of eight colors. The first color (color 0) will be the top color on the left column and the last color (color 31) will be the bottom color on the far right column.

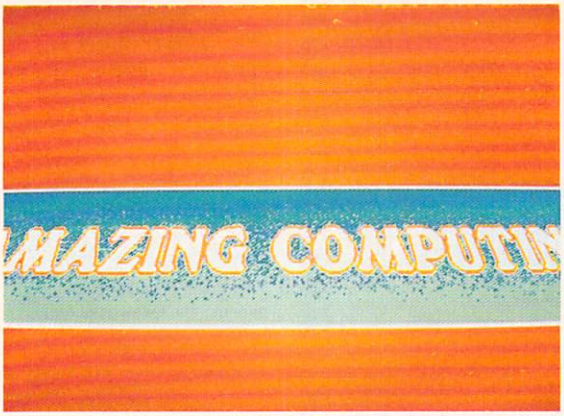
Pull down the Anim menu and go to the Set Frame Count requester. Set the frame count to 30. Click OK. Now go to the last frame. You can do this by either by hitting the number three (which brings up the Go To Frame # requester), entering 30 and clicking OK; or take the shortcut. Just hit "shift-2". This automatically takes you to the last frame in any animation. ("Shift-1" always takes you to the first frame. Learning these keyboard shortcuts can save you a lot of time and effort.) Use the right mouse button on the Font Tool Box to bring up the font requester. Pick the Diamond font, size 20 and then pick bold. (We will assume from this point on that you know to close a requester by clicking OK when you are done.) Make sure that red is your background color, pick the white as your foreground



I'll animate this, you animate your name!







white as your foreground color and then type your name anywhere on the screen, but do it all in caps (capital letters) and put a space between each letter. Pick the 2-pixel square brush (second row from the top-far right) from the brush shapes menu and the unfilled rectangle tool and draw a rectangle about twice as high as your name and an inch or so longer on either side. Go to the palette requester again ("p") and make colors two and three (the top and bottom on the right) a "Range" by first clicking the top color, then Range (the word "to" will appear on the pointer), then the bottom color. Close the requester and bring up the Fill Type requester by clicking the Fill Tool with the right mouse button. Click the up-and-down gradient box then close. Now pick the blue as your foreground color, the Fill as your tool, and fill the box you just made. See how the two colors shade from top to bottom and are dithered in the center?

Now pick the Brush Selector, and using the right button on the mouse pick up your name as a brush. Make sure that your background color is red so that you will pick up only the letters. Move your name so that it is positioned over the rectangle but do not stamp it down just yet. Pick red as your foreground color and hit 'o' on the keyboard. This will outline your name in the current foreground color (red) approximately one pixel thick. Hit 'o' two more times. Now you have a nice contrasting outline around your name. Center your name over the rectangle and stamp it down.

Centering items on a page can be made very simple with the Coordinates feature which I mentioned earlier. I always turn this on before I begin any project. When you pick up something as a brush try to draw the Brush Selector lines evenly around the object. In this case, if the letters of your name are centered within the brush you cut out then you can stamp them down (reading the coordinates on the Title Bar) in the center of the page. In this example, the screen is 704 pixels wide so we know that the center is around 351 (pixels read 0-703) pixels! This is horizontal centering. If I am centering a single object or line of text vertically on a page, I like to keep it just a little above the actual center. This seems to be more pleasing to the eye.

Try picking up the entire rectangle with your name inside as a brush (Brush Selector-right mouse button). Leave about one quarter of an inch around all four sides of the rectangle (for centering) when you cut the brush. Now stamp this centered brush at 351 X 260 (slightly higher than the middle-240).

Before you do anything else, click on Undo. All of your hard work will disappear. We do this because we don't want to draw this brush yet, we just want to tell the Move requester where we want the brush to end up. Even though we can't see the image anymore, the Move requester remembers the brush and where we want it. Now open the Move requester (Move under the Anim menu or "shift-m". Click "Brush" and "Cycle" to de-select them. Next do the following things:

1. Click "Move To". This is because we have given DPaint III the position we want the brush to move to (as opposed to where we want the move to begin).

2. Change the Count to 29. Since we are working backwards (Moving To frame 30) this will leave a blank frame at the beginning. The default setting on Count will always be the actual # of frames you have set so always subtract one frame if you want to start from black.

3. Set "Ease In" to five. This will make sure that there is very

little change in position in the last five frames, so that the brush will "ease" into position. The movement will thus appear smoother.

We want the brush to appear to come from behind the viewer so set the "Z" figure to 500. Now click "Preview". This will show you the moving outline of your brush without actually drawing the frames. This is DPaint III's version of an animator's "pencil test". The frames will flash by fairly fast, but it will be obvious that you have not seen 29 frames. This is because we have set the "Z" too high. The animation now starts so far behind the viewer that the first ten or so frames can't even be seen. These are wasted frames. Reset "Z" to 300 and try Preview again. This begins the move just out of our line of sight. Looks good.

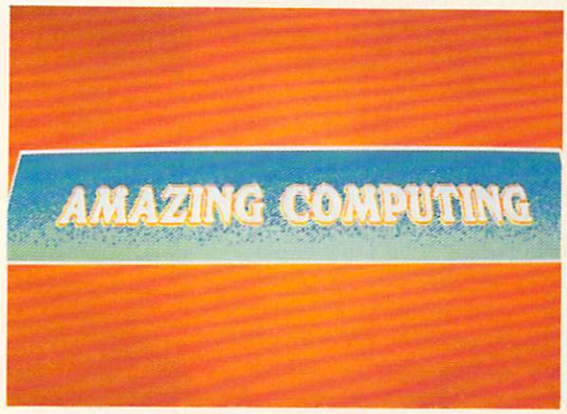
Now set "Y" to 200 so the brush will appear to come from below the viewer. Try Preview. Again we seem to have several wasted frames. Try "Y" at 105. This time the preview looks alright. Just remember: these numbers are set by trial and error. Anytime that you want to have a brush move in from off-screen watch the preview carefully. If more than a couple of seconds go by between the time you click Preview and the time you see your brush moving then you are wasting frames. This doesn't hurt anything, but why have your Amiga waste time trying to draw frames that no one will ever see?

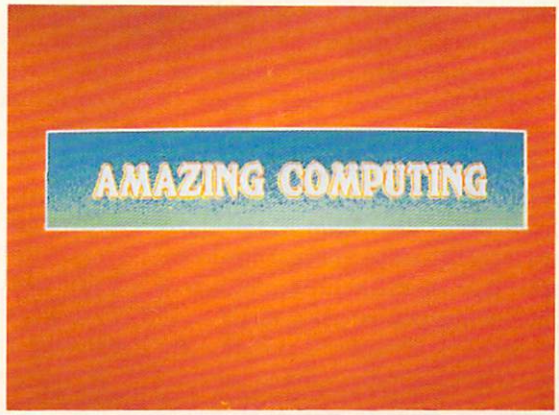
Now we are going to try one more adjustment to make this a really cool move. Set the "X" angle to 90. This will begin your move with the brush at a 90 degree angle to the screen. Remember, by setting "Move To" at the beginning, we have told the program that all these other coordinates are where we want the brush to move from. Now try Preview. "Kinda like Star Wars, huh?" Except that our brush straightens out at the end of its move so that it will be more legible. This will not be terribly impressive in Preview mode as it is a little hard to see, but now that we are through you can click "Draw". You can also go get a cup of coffee (or the refreshment of your choice) as this will take 5-10 minutes.

When you come back from your break, check to see if your animation is still "cooking". If your pointer isn't snoozing you're all done. Now hit F10 to remove the Toolbox and Title Bar and put the screen in overscan mode. Next, hit five to play your animation and voila! There you are. Amazing what you've learned isn't it? How about one more little adjustment? Let's make your name remain on the screen a little longer at the end. Go to the last frame ("shift-2"). Now go to the Set Frame Count Requester (under Anim). Set the frame count to 100. This will add another 70 frames to your animation. It's important to note that frames are added at the point you are in the animation and that they will be identical to the frame you were on when you added them. That is why we went to the last frame. We have now added 70 frames at the end of your animation with your name in the same place for all to admire! Just one final touch now. Make sure you're on the last frame (shift-2, remember?) and hit clear. The Clear requester will pop up asking if you want to clear all frames, a range of frames or just the current frame. Click Current Frame. Now your animation begins and ends with a blank screen. Hit five to play it. Classy, huh? You can save this incredible first animation on that formatted disk to show your friends and family later (at length, no doubt) by going to Save under the Anim menu, naming it and clicking OK.

Now, I will take a couple of questions from the class. You there, in the back. Why did we use overscan even though the background was black? Good question. We used overscan because







our brush was moved in from outside the edge of the screen. If we hadn't used this feature, our brush wouldn't have appeared until it was about one half of an inch above the bottom of the screen. Always remember, even when your background is black, you will need to use overscan (for the best effect and always if your outputting to video) if anything on the screen will be moving off of or onto the screen in any direction.

OK, you, in front. Why do the letters on the name seem to have slightly jagged edges as the brush moves on screen? Another good question. You see, what we did with the move requester was really an automatically generated extension of DeluxePaint III's Perspective feature. DPaint III did the calculations and adjustments to make the image appear as if it was moving away from us and turning in perspective as it moved. Life being what it is (and DPaint III being what it is), nothing is perfect. There is always some distortion when computing perspective moves. This results in the jaggies. There are however, a couple of saving graces to this situation.

First of all, we started with a small brush and had the computer calculate a larger one. I did this for the sake of clarity in this tutorial. Whenever possible, images will be clearer if you start with a large image and have it shrink instead of vice versa. Secondly, you have now seen this move several times. That's why you are so aware of this. It is important to realize that the entire move only takes three or four seconds and by the time the eye focuses on the letters, the jaggies are almost gone. Check this out yourself at home. Watch the commercials very carefully on TV tonight. If you look closely, you will see lots of jaggies in the computer generated titles and such, but they generally go by so quickly that no one notices. Lastly, the DPaint III folks have given us partial relief with an anti-aliasing feature. Anti-aliasing is the technical term for smoothing out the jaggies. Pull down the Settings requester under the Perspective menu and you will see three Anti-aliasing settings. None (the default-which we just used), Low, and High. The reason that None is the default and that you are given choices at all is that there is a trade off in time for the Anti-aliasing feature. If you set Anti-aliasing to High and repeat the tutorial you will find that that first thirty frames will take hours to draw instead of five or ten minutes. While the Anti-aliasing is still not perfect, even on High, the difference is remarkable. For professionals and perfectionists the best solution is to use None (the default), generate your animation, and, if the movement is correct and to your liking, redo the entire thing with the Anti-aliasing set to High while you go out to dinner! Naturally, it will be very simple to repeat the project because we saved all of our brushes to disk and made copious notes of everything thing we did: what we did on what frame, all the figures we gave the move requester, where we stamped the brushes, etc. Right, class?

Bitplanes/Colors Table

1 bitplane	2 bitplanes	3 bitplanes
0	00	000
1	01	001
	10	010
	11	011
		100
		101
		110
		111

As you can see, 1 bitplane allows for two possible colors; 2 bitplanes allow for four possible colors; 3 bitplanes allow for eight colors, etc.

Table One

I think we have time for one more question. Why did we start with only a four color screen? This is a bit complicated but I will make it as non-technical as I can. There are certain hardware limitations to animation speed even on the Amiga. Remember, for every frame of your animation your Amiga has to redraw every single pixel that changes. Now, on the average, there are not that many pixels that change from frame to frame, but the potential is incredible. The screen size of the tutorial we just did was 704 X 480. If my calculator is correct that's 337,920 pixels. If the entire picture changed from one frame to the next that would mean over 300,000 pixels would have to be redrawn. That part is easy to understand. The second factor is how much of a change there is in each pixel. This can get a little confusing.

You must remember that computers deal with information in binary form. This means ones and zeros. If you set your screen for, say, eight colors, that is three bit planes. That is like having three rows of ones and zeros. Four colors is two bitplanes-two rows and two colors is one bitplane-one row.

Look at the one bitplane list (Table One). There are only two possible combinations and, therefore, only two possible colors. To change a pixel from one of these colors to another (no matter what colors they are) requires only one bit to be changed. In the two bitplane example, there are four possible combinations of one and zero, so there are four colors. The biggest change possible is between color zero (the first) and color three (the fourth, or last). This would require changing both bits. As you can see in the three bitplane mode the largest change you can have is all three bits. This is more difficult and time-consuming for the computer than changing one or two bits. Granted, we are not talking large amounts of time here but when you multiply them by a possible 300,000+ it does make a real difference.

Try repeating the tutorial (use no anti-aliasing) in eight color mode (three bitplanes). Use color zero as background, color one for the letters and the outline of the box and use colors six and seven (the last two for the fill. You will find that the animation slows down considerably. You can certainly imagine what this would be like using colors zero, one, fourteen, and fifteen in the full sixteen color (four bitplane) mode!

The trick, therefore, is to always plan your animations thoroughly in advance. Use as few bitplanes as you can get away with and use your colors wisely. If you are moving several colors against a solid background (color zero) make the largest object or the one that moves the most color 1. Always arrange your palette so that colors that change will be as close together as possible.

Well class, it's been another interesting session. I hope it's been as much fun for you as it has for me. Until the next time...

•AC•

Animation with Sculpt-Animate 4D

AMIGA

How To Get Moving In 4D!

by Lonnie Watson

Ever since the Amiga was introduced, people have been trying to animate with it. In the beginning, we thought the animations being produced were great because we finally had a machine that could do real-time animations. But just by looking at them, one could tell that the Amiga was capable of doing so much more. Part of the reason for this shortcoming was the fact that, in the beginning, there was very little software available to help the budding animator in his or her quest for the "Most Awesome Spectacle". In late 1985 and early 1986, the only works of animated art being produced came from programmers. These people possessed the knowledge, cunning, foresight, and perhaps stubbornness that allowed them to program the Amiga.

Now, of course, the situation is different. There are many programs out there that allow animation of various sorts. These animation programs fall roughly into two categories. The page-flipping type is perhaps the most well-known. An example of this type of program are the animation capabilities of DeluxePaint III. Anybody who has seen a master of Amiga art like Jim Sachs perform their magic can attest to the fact that DPaint III is a very powerful program. But there are other animation programs out there in the vast expanse of Amiga software. Programs like Videoscape 3D (and its associated software), PageFlipper, Animagic, and a whole host of others. But perhaps the most powerful of the more affordable software pieces is Sculpt 4D.

Sculpt-Animate 4D (from now on we'll call it Sculpt), is a very powerful program. It is unique in many of its abilities and has a lot to offer. This article will not try to be a Sculpt tutor. To give a detailed tutorial on the effective use of Sculpt would create a tome larger than most phone books! That is not to say that Sculpt is a very difficult program to use, just that its extreme versatility makes for a very complex program.

Sculpt's animation abilities fall into two different modes. The Sculpt manual separates them into Global and Frame methods. Global animation is a little difficult to grasp at first. It helped me to think of Global animation as a movie set, with myself as the director. To work in the Global animation mode, you have to create an action for each piece of the picture. Each element's action must be completely thought-out in advance. You then instruct each element as to what to do and yell "ACTION!" With a fair degree of forethought, you can create some very stunning effects this way. In fact, some of the "plane-flying-in-a-circle" animations can only be done effectively with Global animation. Setting up a Global animation is not really that hard.

An animation project in Sculpt is known as a Take. The first step in creating an animation in Sculpt using Global Mode requires that you specify the name of a Take by executing the PROJECT/LOAD/TAKE pull-down menu. You will then be instructed to enter the name of the Take that you want to work on. Selection of an existing Take will open that Take up and allow you to edit its characteristics. If, as in this case, you are starting a new Take, enter its name manually in the file box and select OK with the mouse. There will be no apparent change in the program's operation, but you will now be able to use all of Sculpt's features to create your imagery. The difference is that Sculpt has now unghosted the EDIT/MODIFY/TAKE menu item. Selection of this menu item will bring up the powerful Modify Take requester.

Creating an actual animation in Global Mode is rather easy in Sculpt. First you need an object to animate. Using the standard tools provided by Sculpt makes this task a breeze. For this excercise, make a very simple object — say, a box. The next step is not quite as easy: you have to create a path for your object to travel on. The need for this should be quite apparent. In order to direct something to move, you have to tell it how and where to move. The path that you want the object to take must be created. Sculpt gives you the tools necessary to make an object path. You simply create the shape of your path with the curve tool or some of the other automatic tools. You must take care not to create a shape that has any forks in it. Examples of suitable shapes would be zig-zag lines, circles, and ovals, etc. In this case, try a large circle The next step requires that you determine the direction of travel along the path (will the box move clockwise or counter-clockwise?) This is done by highlighting the first point on the path (your start point), then placing the crosshair cursor on the next adjacent point in the desired direction. This direction is now specified as the direction of travel. Now that you have specified the shape of your path and the direction of travel, you have to actually name it as a path. This step tells Sculpt that this shape you have created is going to be a path, and it is done by selecting the EDIT/DO/MAKE/PATH pull-down menu. Now Sculpt knows that this shape is a Path.

Global animation is a little difficult to grasp at first. It helped me to think of Global animation as a movie set, with myself as the director.

The next step is to name the Path. By selecting a point on the Path and then selecting the EDIT/SELECT/CONNECTED pull-down menu, you will highlight all the vertices on the path. Then, if you select the EDIT/NAME/SELECTED VERTICES pull-down, you will get the NAME VERTICES requester. Sculpt allows you to group points together and give them a name (for convenience). Once the group of points is named, you can treat them, in a way, as one unit. (In this example we are going to animate a box. We do want to make sure that all the points in the box move together, don't we? If we group the points together by naming them, then all the points associated with that name will move together.) In this instance, we want to name the Path. Clicking in the Add Name box, you can type in the name that you want. Pressing Return will place that name in the list of names maintained by Sculpt. Click OK to leave this requester.

Now we go back to the object we created. Select one point on the object and select the EDIT/SELECT/CONNECTED pull-down. This will select all the vertices in the object. Then selecting the EDIT/NAME/SELECTED VERTICES again will bring up the Name Vertices requester once more. You should see that it already has the name of your Path in the list of names. Clicking on the offspring box will move the entered name of the Path to the Parentage box and clear the list. Now enter the name for the object in the add name box and press Return. This will place the new name in the list of names.

We have really done two things at once here. First, we have named the object (as we previously named the Path). Second, we set the Path up as a Parent and Box as an Offspring. This is what Sculpt calls Hierarchy. This is a very valuable tool in Sculpt as it allows you to set up conditions wherein, if one object in a list moves, all the offspring of that object move with it. This is called Hierarchical Motion. For instance, suppose you had a model of a plane, and on this model of the plane you had a propeller and landing gear. Now, a real plane takes its propeller and landing gear with it when it moves (at least, that is the theory). Sculpt allows you to do the same thing by creating a hierarchy. Then, if you move the body of the plane, the associated propeller and landing gear move with it.

The next step is to tie the object to the Path. First you need to highlight the entire object by selecting a point on the object and then selecting the EDIT/SELECT/CONNECTED pull-down menu. This will highlight each of the object's vertices. Then, place the cursor on the object's center of gravity. This may seem confusing but the center of gravity will not necessarily be inside the object. For instance, a door will have its center of gravity along one edge.This step can be the key to creating realistic-looking animation. After selecting the center of gravity, you need to select the EDIT/MODIFY/LOCAL ORIGIN pull-down. Selecting the Set button will tie the object to the path at the center of gravity. Once you have done this, you can place any light sources that you want, as well as place the observer and target.

Selecting the EDIT/MODIFY/TAKE pull-down once again will present you with the Modify Take Requester once again. Clicking on the Number of Frames Box will delete the number already in the Box. Entering the number of points on the path that you have created will create an animation with as

many frames as there are points in your path. You can then click on the Save Global Scene button to save the animation to your disk as a take. Remember, this is the raw data of the Take — not the finished, rendered Take. Once it is saved to disk, you can then load it back at any time and edit it. Selecting the Preview button will allow you to view the animation in wireframe. This will give you a good feel for the animation and help you see problems with your work before you wait five days for the whole ray-traced animation to finish.

As we have seen here, the Global animation feature of Sculpt is quite impressive. It allows you to create a path for an object and make that object move along that pre-defined path. It is important to remember one thing about this type of movement. Each point along the path that the object is tied to represents its position from frame to frame. If you have an object moving in a straight line and the only points on that line path are at the paths ends, then the object will zip from one end of the path to the other in two frames. If that path is long, then the animated effect will not be that good. If, on the other hand you had many points on that straight path, then the effect would be slower and smoother, and overall quite good. It is also important to make the distances between the points on your path equal. Otherwise the object will appear to speed up and slow down as it travels on its path. This effect can be quite good (and even necessary), but only when desired.

Sculpt has another mode of animation called Key Frame mode. Key Frame mode works on a simpler principle than the Global mode. Key Frame mode creates animation by a process called "tweening". Basically, you create animations by selecting two frames as a reference. In each of these frames, you have some objects that have moved. Let's say a ball is in the air on the first frame, and on the ground in the second. These frames are called key frames. Sculpt is then able to create the "in between" frames so that the overall motion of the ball is smooth during its fall to the ground. All you have to do is tell Sculpt how many frames to render between the key frames.

Key Frame animation is accomplished via the EDIT/MODIFY/TAKE requester once again. When it appears, you select the Global gadget on the top and it changes to the word Frame. From here, you will be able to edit and create key frames for your animation. In the middle of the requester, you will see an object that looks like a piece of film. There will be an arrow pointing to the current frame in that filmstrip. Clicking on the Key Frame button will toggle the frame status from Key Frame to non-Key Frame. Selection of the Save Key Frame button will save the current image in the Tri-View to that Take's Key Frame data. You can then exit this requester and move the wanted objects around a bit, re-enter the Edit Take Requester, select another frame, make it a Key Frame, and finally save it to the Take's file by selecting the Save Key Frame scene button again.

The Global Mode of the Modify Take Requester has certain effects on the Frame Mode on the same requester. For instance, the number of frames in the Global Mode of the Modify Take Requester sets the length of the filmstrip in both modes. You also need to go back to Global Mode to Preview the animation or to Render it in its entirety. You may well wonder why this is the case. At first I wondered why could they not have just made the two requesters separate and allowed each to have completely different settings? Well after playing for a while, I think I know why. Global and Frame Modes are really the same thing. Global mode simply allows some different ways of defining motion. You can have Global and Frame animation occurring in the same take. This could be useful — for example, if you had straight linear motion on some items and more complex motion on other items.

Key Frame mode works on a simpler principle than the Global mode, creating animation by a process called 'tweening'.

There are a few things to keep in mind when doing Frame animation. The first is the effect of the linear interpolation that Sculpt performs when doing frame animation. For example, suppose you had a rectangle sitting on the ground, and you made the second key frame with the rectangle rotated 90 degrees or so. The animation would create some frames with a rectangle that is very different-looking than the beginning and ending rectangles. (It would most likely be much much smaller). This is an inherent problem with linear interpolation, not Sculpt specifically. The solution here is to split the single tween up into many smaller tweens, or to tie the rectangle to a path (Global Animation) rather than using Frame Mode. Secondly, Frame Animation is not really suitable for complex movements of objects. This is because the time required to create each key position in a Frame Mode anim is far greater than simply tying the same objects to a path and using Global animation.

Keeping in mind the differences of these two powerful modes of animation in Sculpt, one can create some truly stunning artwork. Sculpt is a very complex and powerful program. I have really not even scratched the surface here. Perhaps after reading this, you will have the desire to explore Sculpt a little more. I am sure you won't be disappointed. •AC•

Sculpt-Animate 4D
Byte by Byte
Arboretum Plaza II 9442
Capitol of Texas Hwy. N., Suite 150
Austin, TX 78759
(512) 343-4357
Price: $499.00
Inquiry # 208

PIC-MAGIC:

A Product Profile

by Orlando Furioso

It looked like it was going to be a fairly busy weekend, but I wanted to check out this new clip art set. I was sure I could spare and hour or two first thing Saturday. Next time I'll open the package before I make any plans.

It was a rather unassuming envelope, although it did seem a bit bulky for just clip art. Once open I discovered a beautiful little fan-file disc case inside which holds the ten (TEN?) discs that make up the set. You may have seen this case before: the hard plastic kind that flips open and sells for around $10 at your local dealer. Very nice. I also found the manual — all 220 pages worth.

THE MANUAL

The manual begins with the usual "thank you's" and credits, followed by a nice letter from Joey Rubenstein, President of Joe's First Company, Inc. It is an introduction and hello from an apparently very down-to-earth guy along with an explanation of the raison d'etre of this particular package. This is followed by several pages of technical explanations and instructions, which are necessary because of the unique format of the set.

Joey's set, which is only the first of many planned, was apparently designed to accommodate not only DeskTop Videographers but DeskTop Publishers as well. Knowing that structured clip art is useless to video and graphics users and that the usual screen-sized bitmapped graphics are frequently not suitable for the publishing users, he has done something very different. He has compiled a package of oversized bitmapped clip art. They are what some call superbitmaps

THE SIZE

The 250 plus images in this package are all over-sized pictures, many as large as four overscanned screens. They are this size because, as we all know, it is much easier (i.e., cleaner) to re-size graphics smaller rather than large. The manual gives clear and concise instructions on re-sizing these pictures for printing and video work. It even explains how to figure out proportions for shrinking an image to a particular size. The large size of the pictures makes them perfect for printing big posters, banners and other related applications.

I found resizing to go relatively smoothly when I tried bringing some of the pictures down to more typical sizes for video work. Joey's instructions for DPaint III suggest loading the picture then going to the screen format requester and changing the picture size to screen (instead of keeping it the same). In some cases, this requires you to resize the resulting image (via the stretch device) to restore the correct proportions. In many cases I found it easier to just cut the picture into smaller brushes, shrinking them individually ('h' in DPaint) and then putting them back together (usually on the spare page). In any case, neither method was terribly difficult.

AMIGA AMIGA AMIGA AMIGA

AMIGA

Brushing-Up On...

by Barry Solomon

There are a lot of people doing animation with their Amigas these days. Some are animating at home and some at work. Some are professional animators and some are not. Many of these people are not even artists. With the ever-increasing popularity of Amiga animation (Amigamation?) many people have become and are becoming animators who would never have thought of themselves as artists.

In the Amiga world of cel animation (drawing one frame at a time) one of the fastest and easiest ways of producing animated objects, backgrounds, people etc. is to use clip-art. It's great to be able to draw a beautiful jet airplane to move across the screen in your animation but if you're not an artist or, if you can draw a bit but planes aren't your thing why not use clip art?

In many animation situations the object we animate only need to move on screen. They don't need to change in any way. Obviously it does depend on the situation and ones of the easiest ways to keep your from looking right is not to have some objects change. Take, for example, a car. If you move a car straight across a screen over several frames it will look very wooden. Of course, if you also move it up and down a tiny bit, or maybe use some streaks to show speed it will look much more natural. Maybe even some very simple color cycling on the wheels or just having the driver's scarf blowing in the breeze! Perhaps a combination of one or more of these techniques will yield just the effect you're looking for, but the basic point is that we can find ways to animate a car without necessarily being either an artist or a mechanical draftsperson.

There are any number of clip art packages available for the Amiga and, with a little practice even a non-artist can learn little tricks to animate objects or even people.

Frankly, one of the best ways to learn economical (i.e. easy) animation is to watch the Saturday morning cartoons. Light-years away from the old Disney-type (or even Warner Bros. type) of animation, the current crop of week-end animations uses every possible trick to economize.

when doing an article of this type, but I must say I fell in love with this package. The images on the ten disks are broken down according to these catagories:

1. Food
2. Sports
3. Cars & Other Wonders
4. Christmas
5. People
6. Plants
7. Animals
8. Miscellaneous Objects
9. Bordersets
10. Eyegrabbers

The assortment of individual clips on each disk is delightful, but it is the style that really caught my eye. Much of the clip art on the Amiga market tries so hard to take advantage of the Amiga's incredible graphics capabilities that it ignores the talents of the artists who wish to use this art. Photographic realism has its place, but often the palettes of such clips are so vital to the picture that it is almost impossible to fit them into your work. This is not the case with PIC-MAGIC. This package is strictly black and white line art obviously sketched with a very relaxed and casual style. The images are very realistic, but they have that 'alive' quality that separates a good drawing from a photo.

Even with the enormous quantity of pictures in this set, the amount of serious thought that went into choosing them is obvious. Going through each image in the manual (yes, all 250+ are in there) the assortment is amazing; certainly one of the best chosen assortments of clip art I have ever seen. A picture can be found for just about every purpose; from illustrating an article to decorating a menu to making posters to support your local teams. Particularly in terms of advertising uses such as posters, flyers etc., I found very little lacking in this set.



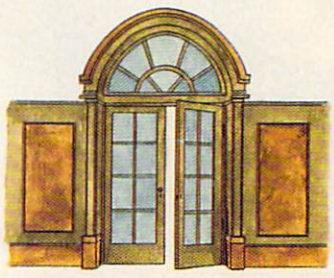
Miss Pineapple*

About the only complaint I could find with this entire package (and it's a small one) is that in terms of video uses, they could have made life a little easier for the extremely lazy. The pictures are so nice it would be great if the sizing step could be eliminated. What I would really like to see is a Package One A which has all 250+ images on just a couple of disks already brought down to a more average size for video work. The funny thing is, when I mentioned this to AC's Video Editor, he had just gotten through saying the same thing to Joey Rubenstein on the phone and Joey thought it was a great idea. Talk about responsive.

According to Joey, PIC-MAGIC #2 is almost done, and soon they will be starting on other packages. It will cost them a little extra, but with any luck they will make all of the packages available in two sizes: one for video and one for print. It's tough to really complain about this package, because at a retail price of $99.95 for the set, each image costs about 40 cents, and at that price, if I have to shrink a picture down a little it's no big deal.

If you're looking for photographic quality clip art, this would be a good package to avoid. However, if you prefer a huge selection of excellently sketched, ready to color, images, this is the package you've been waiting for!

•AC•



Glass Doors*

Just a quick P.S.

I was speaking to Joey Rubenstein just before this issue went to press and he informed me that he has now made this a ***twelve-disk set****. The extra two disks will contain approximately 98% of the 250 pictures scaled down to a more usable size for video. Registered owners will receive the upgrade automatically. Should you purchase the ten-disk set from a dealer's current stock, you will receive the upgrade when you send in the registration.* ***O.F.***

*Images have been colored for enhancement.

PIC-MAGIC
Professional Quality Clip Art
Package One
Joe's First Company Inc.
P.O. Box 579, Station Z
Toronto, Ontario,
Canada M5N2Z6
(416) 322-6119
Price, $99.95
Inquiry # 207

Animation!

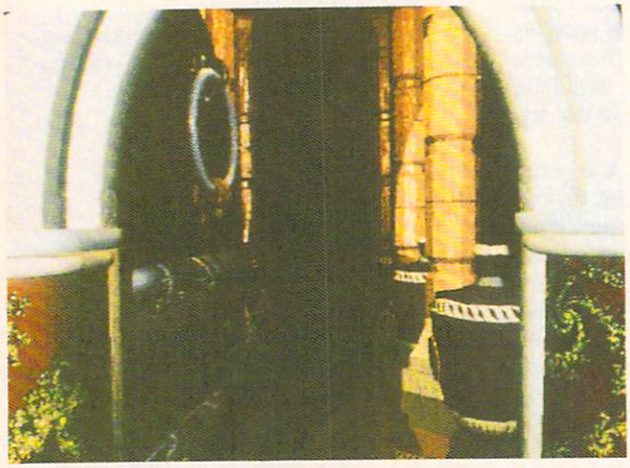
Rarely do you see a character running into a wall. What you usually see now is that character running off the screen, followed by a bang or other noise while the screen shakes followed by a cut to the injured character. These are exactly the kind of shortcuts that you can learn to use in your animations. If you use them wisely (and as long as you are not trying to compete with Disney) you can make your animating easier and give it a more professional look.

Even clip art animals and people can be animated by careful cut-&-paste methods. Rotation of brushes can be difficult as there is no such thing as perfect rotation (unless you're working in the structured art/postscript area) and your brush will likely suffer serious distortion as well as a serious case of the jaggies. But, even rotation can be done if you keep it to a minimum and you're willing to do some cleanup.

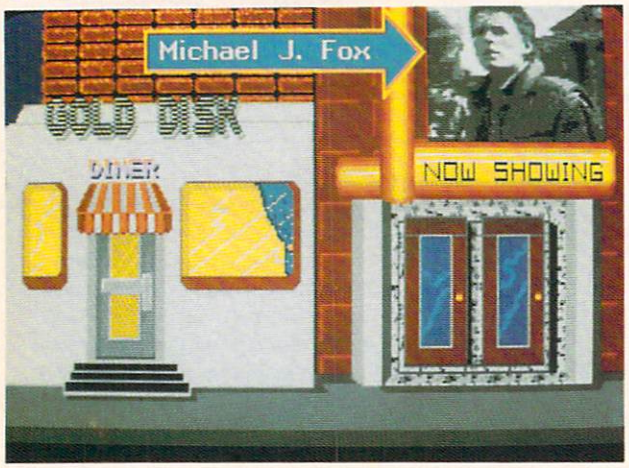
One note about animating clip art. there is one package of clip art on the market called Build 'Em from The Picturebox. Build 'Em is a figure construction set containing 25 screens with over 500 body parts (sounds like a Wes Craven film) which are anti-aliased and designed to be cut and rotated.

There are a lot of Amiga clip art packages available and with the success of the mega-set of clip art from Joe's First Company the future of Amiga clip art looks bright. So with a little practice you might just be able to cut, paste and animate your way to fame!

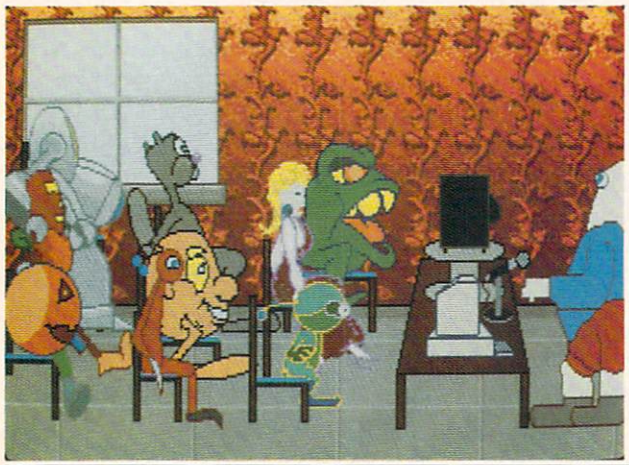




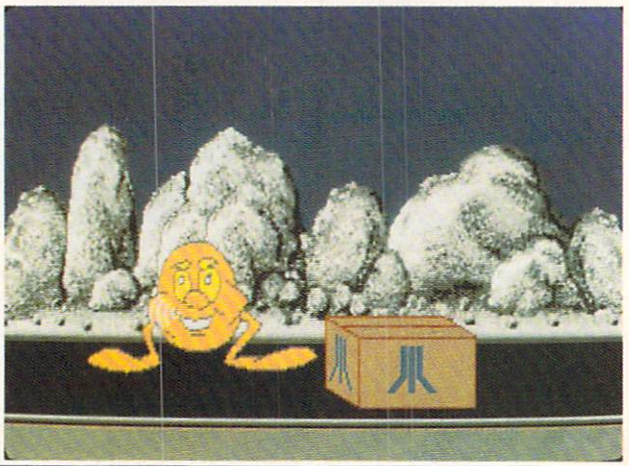
Above and Left: **The Sentinel,** *by Bradley Schenck, won first place in this year's BKDC.*



Left and Below Left: Roger Curren's **VAUX-Killer** *entry in the BKDC won him second place.*



Right: The Atari box is about to be smashed in the **VAUX-Killer** *animation.*



PD Serendipity

Insight into the World of Freely Redistributable Software

by Michael Morrison

The Fred Fish collection has made it to Number 282! Mr. Fish must be very busy. As always, there are many useful programs in this latest release, but many of the BADGE Killer Demo Contest entries were also included. And since we do have an animation theme going this month, I decided to give these people some credit for their work and present parts of their animations for the world to see. This year's winner requires at least 3 meg to run and even has a 3.5 meg version!

THE SENTINEL

This animation was created using Turbo Silver, Deluxe Photolab, DeluxePaint III, and The Director. The animation consists of about 70 frames of ray-traced pictures. It took approximately three and a half weeks to render the animation.

The Sentinel was entered by Bradley Schenck (see related article, "The Sentinel", on page 42). Because of its size, the animation had to be put on two disks, Fred Fish disks #264 and #265. Bradley won first place with this animation.

VAUX-KILLER

This animation was entered by Roger Curren. It is available on Fred Fish disks #265 and #266. The animation uses a clever approach—sort of a contest entry within a contest entry, as it tells of anxieties encountered while partaking in the VAUX-Killer contest. It is quite long and has some very funny parts. Roger won second place with this animation.

The AMIGA 1989 BADGE Killer Demo Contest

SPIGOT

This animation was entered by Eric J. Fleischer (Dr. Gandalf). The animation consists of many ray-traced HAM overscan pictures with sound effects. Eric won sixth place with this entry. It appears on Fred Fish disk #272.

TOO MUCH 3D

This animation was entered by Jim Robinson. There isn't any information on how Jim made the animation. *Too Much 3D* shows what happens if you spend too much time doing 3D work on your Amiga. Jim received tenth place for this entry, which is on Fred Fish disk #270.

CPU STANDOFF

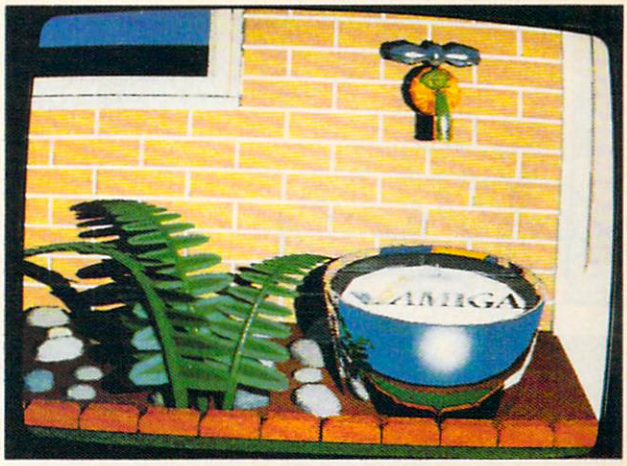
This animation is a visual representation of the computer wars. It has some interesting scenes with nice art. There are many visual jokes aimed at other computers throughout the animation. *CPU Standoff* was entered by Bob Janousek and is on Fred Fish disk #271.

BALLOON

This animation was entered by Dave Schreiber. It appears to have been ray traced and includes sound effects. *Balloon* is on Fred Fish disk #272

LASER BOING

This animation shows a ray-traced Boing ball spinning around with laser light bouncing off it. Has a nice effect and also includes sound. This animation was entered by Mike van der Sommen and is on Fred Fish disk #271. •AC•

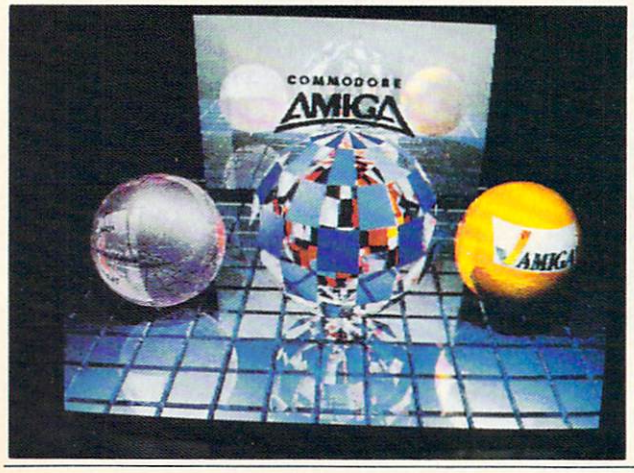


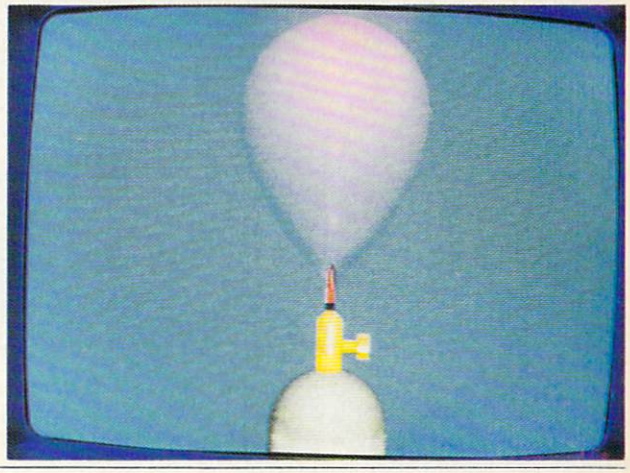


Above Right: Eric Fleischer's (Dr. Gandalf) entry in the BKDC won him sixth place. Right: **Too Much 3D** ***by Jim Robinson won tenth place in the BKDC.***



Right: Bob Janousek's **CPU Standoff** ***shows the ongoing battle for computer supremacy. Below Right: Dave Schreiber's*** **Balloon** ***animation shows a balloon filling up and then flying through the air.***





Left: **Laser Boing** ***by Mike van der Sommen showed a Boing ball energized by laser light.***

AMIGA

Animation? Basically!

Cell Animation in AmigaBASIC

by Mike Morrison

There are many different ways to produce animation on the Amiga. The operating system has many fancy structures, functions, and commands to help make animation on the Amiga easy. This array of tools probably does make animation easier if you are programming in a compiled language. But what about AmigaBASIC?

ANIMATION IN AMIGABASIC?

You could take advantage of every one of these structures, functions, and commands by using the AmigaBASIC LIBRARY and DECLARE FUNCTION commands. Toss in a few hours of skimming through the ROM Kernel Manuals, old Amazing Computings, and the Enquirer (just to relieve stress) and you can end up with a nice animation. But why do all that when you can use standard AmigaBASIC commands and a technique called "cell animation".

CELL ANIMATION

The best example to explain the theory behind cell animation is to turn back the clock to when we where in grade school. Everybody has taking a stack of that gray recycled math paper and drawn a stick figure slightly changed on each sheet. Then when you flip through the stack of paper, wa-la!, animation. Eat your heart out Walt Disney!

USING TOOLS

Because I am inherently lazy, I've learned to use the tools that are available to me. We are going to use many tools that the average Amiga user has. Two of these tools you definitely have: AmigaBASIC, and a program called 'LoadILBM-SaveACBM'. AmigaBASIC is on your Extras disk that came with your Amiga. So is the 'LoadILBM-SaveACBM' program. It is located in the "BasicDemos" drawer. The third tool is your favorite drawing program. I will be using DeluxePaint III.

GETTING STARTED

The first step in our project is to decide what you want to animate. Initially you should keep it simple. Then when you feel comfortable with the procedure, you can let your imagination go wild (within the constraints of the law, of course). I decided it would be fun to start with a deflated Boing Ball and then have it blow up.

Fire up your drawing program. Select a lo-res screen with four colors. Draw a box with the box tool that is 45 pixels square. Then cut and stamp six of these in a row (see Picture One). Draw your animation one frame at a time. Change each frame slightly from left to right. The leftmost frame will be the first frame in the animation and the right frame will be the final frame in the animation (see photo two). Save the picture as CellAnim.IFF. Then exit your drawing program.

Start AmigaBASIC by double-clicking the icon on the Extras disk. Click the mouse in the left of the two windows that appear. The right window is the listing window and the left is the command or immediate mode window. Type:

```
load "df1:BasicDemos/LoadILBM-SaveACBM"
<return>
```

Change the drive in these examples if the file is on a disk in a different drive. After a few seconds the prompt will return. Type:

```
run <return>
```

There will be some text that appears and then you will be asked:

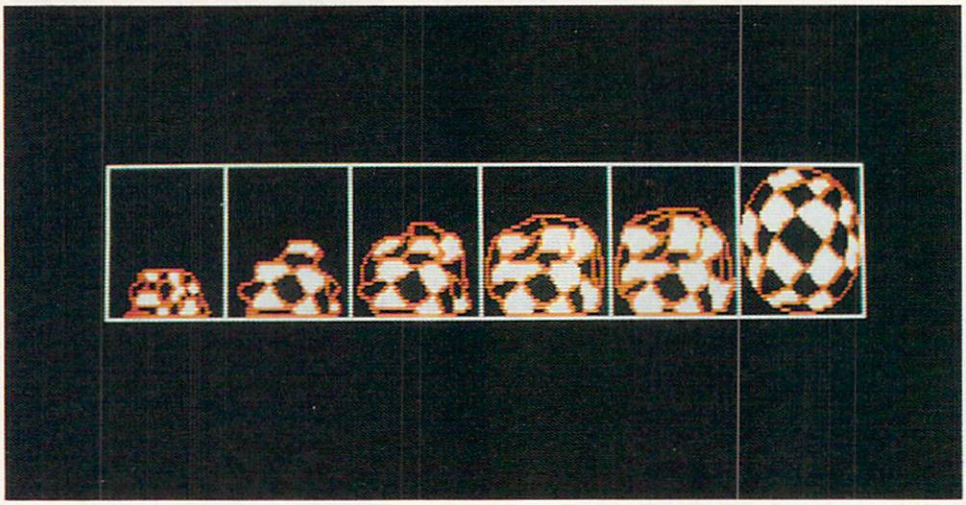
```
'IFF ILBM
filespec ?'
```

Type:

```
df0:CellAnim.iff
<return>
```

You will then be asked:

```
ACBM filespec ?
```



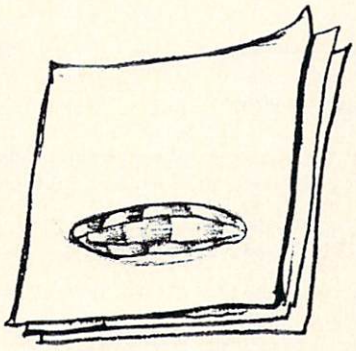
This picture in DeluxePaint shows how the animation looks cell-by-cell.

Type:

```
'df0:CellAnim.acbm' <return>
```

The picture will be loaded in (very slowly) and then the picture will be saved in ACBM format. The ACBM format can be loaded into AmigaBASIC quickly using a program called 'Load-ACBM'. While the program is loading the IFF picture and saving it in ACBM format do not touch the mouse button as this will mess up the picture.

MODIFYING LOADACBM



When the picture is done being saved as an ACBM, click in the left window and type in the program in Listing One. Save it just in case we accidently lose it in this next part. Highlight the entire listing by clicking the mouse at the top of the listing and dragging it to the end of the listing. The whole listing should be orange. Use the pull down 'edit' menu and select the 'cut' option. I know you're thinking this sounds crazy, but trust me. If things seem confusing, refer to Listing Two, which is the entire modified listing.

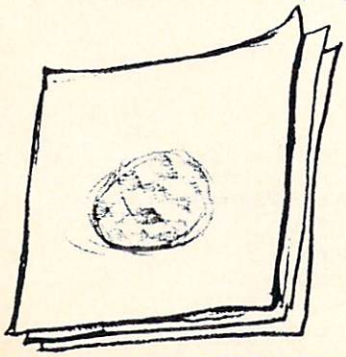
The list window should be empty, with Listing One hidden away in the cut and paste buffer. Type the following in the left window:

```
load "LoadACBM" <return>
```

When the cursor returns select 'show list' from the 'Windows' menu. Click the mouse inside the list window and press the 'Alt' key and the 'down arrow' key at the same time. This should put the cursor at the end of the listing. Select the 'Paste' option from the 'Edit' menu. There is Listing One! We are going to use the LoadACBM program to display our AnimPic.ACBM picture so we can do some work on it. There are still a few changes we have to make to LoadACBM first. Press the 'Alt' key and the 'up arrow' key at the same time. This gets you to the top of the listing. Type in the following command:

```
clear ,19000
```

This increases the amount of memory that AmigaBASIC has for its variables. When we use the GET command it requires a good size chunk of memory to store each cell of the animation. This number was not derived by any known formula—I kept increasing this number untill I stopped getting 'Out of memory' errors.



Press the 'shift' key and the 'down arrow' key four times. At the bottom of the screen you should see the line:

```
input "   ACBM filespec";
ACBMname$
```

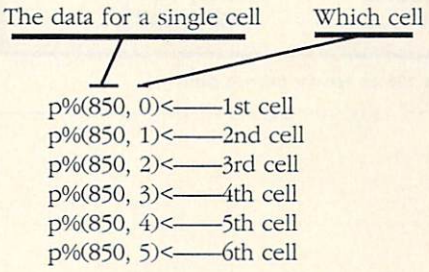
delete this line and replace it with the following line:

```
ACBMname$="df0:CellAnim.ACBM"
```

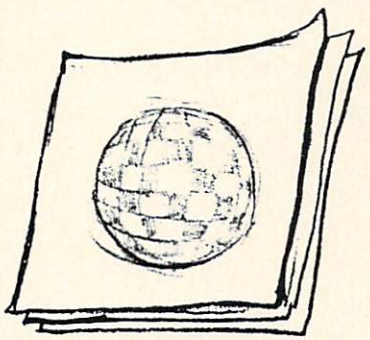
That is all the required modifications. If you want you can delete all the lines that have REMs and PRINTs. These are not really needed anymore and just cause clutter (leave all copyrights of coarse). It is now time to run the program by selecting the 'start' option from the 'run' menu. The picture should be displayed on the screen. Follow the instructions for clicking the coordinates of each cell into the computer. The program uses these to calculate the location of each cell so it can GET them and then later animate them with the PUT command. This can be inaccurate, so the animation might be off slightly. This could be improved by replacing this section of code with the actual x and y coordinates of the cells. The paint program that you used can probably help by displaying the coordinates as you draw each cell.This would produce an exact placement of each cell in the animation.

THE CODE

Listing One is plain AmigaBASIC. The first part of the code gets the coordinates of each cell. This information is used in conjunction with the GET command to capture each of the six cells into an array called p%. This is a two dimensional array. The first index of the array is the area where the data of each cell is stored. The second index is the actual number of the cell. Explanation:



When the data has been copied off of the screen into the array we can then clear the screen with the CLS command. The FOR/NEXT loop then plays through the cells one at a time in the same location using the PUT command. Then for fun I play the loop backwards. This happens five times and then the program ends.



This might seem to be a lot of work for a little animation, but after you have gone through the steps a few times you will see that it really is not that hard. The next thing you should add to this program is a routine that saves the array to disk. This way, once you have the data saved to disk, you could get rid of the whole LoadACBM program and just load the animation data from disk into an array and play the animation without all that overhead. The other area that could use some work is the routine that finds the coordinates for each cell. This could be made a stand-alone utility. Have fun.

Listings begin on following page.

The Bit Bucket

COMPUTER STORE

One of the Oldest and Largest AMIGA dealers in New England

Call us for competitive quotes on the Latest AMIGA Desktop Video products

SPECIALISTS IN AMIGA DESKTOP VIDEO

We offer Desktop Video Training for the latest Amiga Hardware & Software

COMMODORE AMIGA™ Authorized Dealer

Authorized Commodore Amiga Dealer and Service Center

Genlocks *Framegrabbers* *Digitizers*

2 Locations to Serve You

1294 Washington St
West Newton, MA 02165
617-964-3080

621 Boston Post Road
Sudbury, MA 01776
508-443-9731

Circle 139 on Reader Service card.

LISTING ONE

```
blowingup:
PALETTE 1,1,1,1
PALETTE 3,1,0,0

DIM p%(850,5)

PRINT "Click on upper-left of Cell 1"
WHILE  MOUSE(0)<>0
WEND                          'Clear the mouse buffer
WHILE  MOUSE(0)=0
  x1=MOUSE(1):y1=MOUSE(2)
WEND

PRINT "Click on lower-right of Cell 1"
WHILE MOUSE(0)<>0
WEND
WHILE  MOUSE(0)=0
 x2=MOUSE(1):y2=MOUSE(2)
WEND
cellwidth=x2-x1

FOR x=x1 TO x1+(cellwidth*5) STEP cellwidth-1
GET   (x+1,y1)-(x+cellwidth-2,y2-1),p%(0,c)
c=c+1
NEXT

CLS
FOR l=1 TO 5
  FOR x=0 TO 5
    PUT (100,100),p%(0,x),PSET
      FOR d=1 TO 500:NEXT
  NEXT
FOR d=1 TO 750:NEXT d

  FOR x=5 TO 0 STEP -1
    PUT (100,100),p%(0,x),PSET
       FOR d=1 TO 500:NEXT
  NEXT
NEXT

WINDOW CLOSE 2
SCREEN CLOSE 2
```

LISTING TWO

```
CLEAR ,50000&

REM This code is based on the LoadACBM program that comes on
REM the Extras disk and was written by Carolyn Scheppner-CBM

Main:

DIM bPlane&(5), cTabWork%(32), cTabSave%(32)

REM - Functions from dos.library
DECLARE FUNCTION xOpen&   LIBRARY
DECLARE FUNCTION xRead&   LIBRARY
DECLARE FUNCTION xWrite&  LIBRARY
REM - xClose returns no value

REM - Functions from exec.library
DECLARE FUNCTION AllocMem&() LIBRARY
REM - FreeMem returns no value

LIBRARY  "dos.library"
LIBRARY  "exec.library"
LIBRARY  "graphics.library"

GetNames:
'INPUT "    ACBM filespec";ACBMname$
acbmname$="df0:blowingup.acbm"

REM - Load the ACBM pic
loadError$ = ""
GOSUB LoadACBM
IF loadError$ <> "" THEN GOTO Mcleanup

REM - Demo Graphicraft color cycling
IF foundCCRT AND ccrtDir% THEN
   REM - Save colors
   FOR kk = 0 TO nColors% -1
      cTabSave%(kk) = PEEKW(colorTab&+(kk*2))
      cTabWork%(kk) = cTabSave%(kk)
   NEXT

   REM - Cycle colors
   FOR kk = 0 TO 80
      IF ccrtDir% = 1 THEN
         GOSUB Fcycle
      ELSE
         GOSUB Bcycle
      END IF

      CALL LoadRGB4&(sViewPort&,VARPTR(cTabWork%(0)),nColors%)
      REM - Delays approximated
      FOR del = 0 TO ccrtSecs& * 3000
         FOR de2 = 0 TO ccrtMics& / 500
         NEXT
```

```
      NEXT
   NEXT

   REM - Restore colors
   CALL LoadRGB4&(sViewPort&,VARPTR(cTabSave%(0)),nColors%)
END IF

Mcleanup:

Mcleanup2:
LIBRARY CLOSE
IF loadError$ <> "" THEN PRINT loadError$
GOTO blowingup
END


Bcycle:   'Backward color cycle
cTemp% = cTabWork%(ccrtEnd%)
FOR jj = ccrtEnd%-1 TO ccrtStart% STEP -1
    cTabWork%(jj+1) = cTabWork%(jj)
NEXT
cTabWork%(ccrtStart%) = cTemp%
RETURN

Fcycle:   'Forward color cycle
cTemp% = cTabWork%(ccrtStart%)
FOR jj = ccrtStart%+1 TO ccrtEnd%
    cTabWork%(jj-1) = cTabWork%(jj)
NEXT
cTabWork%(ccrtEnd%) = cTemp%
RETURN


LoadACBM:
REM - Requires the following variables
REM - to have been initialized:
REM -     ACBMname$ (ACBM filespec)

REM - init variables
f$ = acbmname$
fHandle& = 0
mybuf& = 0
foundBMHD = 0
foundCMAP = 0
foundCamg = 0
foundCCRT = 0
foundABIT = 0

REM - From include/libraries/dos.h
REM - MODE_NEWFILE = 1006
REM - MODE_OLDFILE = 1005

filename$ = f$ + CHR$(0)
fHandle& = xOpen&(SADD(filename$),1005)
IF fHandle& = 0 THEN
   loadError$ = "Can't open/find pic file"
   GOTO Lcleanup
END IF


REM - Alloc ram for work buffers
ClearPublic& = 65537&
mybufsize& = 360
mybuf& = AllocMem&(mybufsize&,ClearPublic&)
IF mybuf& = 0 THEN
   loadError$ = "Can't alloc buffer"
```


DPAINT III
(PLUS MOVIESETTER USERS)
ANIMATED FONTS
Bring your screens to life with
3D FONT--A full rotation 3D font
For effects that will knock their socks off !!
DISSOLVE FONT--Yes it does !
Dissolve on or off screen - Rotate, Shrink, etc.
POUR FONT-Pour in place WOW
Animated paint can pours the font on screen !
COMIC FONT-See to believe !
Animated characters that bring your title to life
OVER 270 ANIMATED BRUSHES
Thousands of screens that bring out your best
ONLY $ 39.95 Delivered to your door !
Check or M.O. to : ANIVISION
Two Disk SET
P.O. Box 801
PROSSER, WA 99350
WA RES ADD $3.12 TX
COMING SOON-PREHISTORIC AND SCI - FI
products named are trademarks of there respective co.
Circle 150 on Reader Service card.


```
   GOTO Lcleanup
END IF

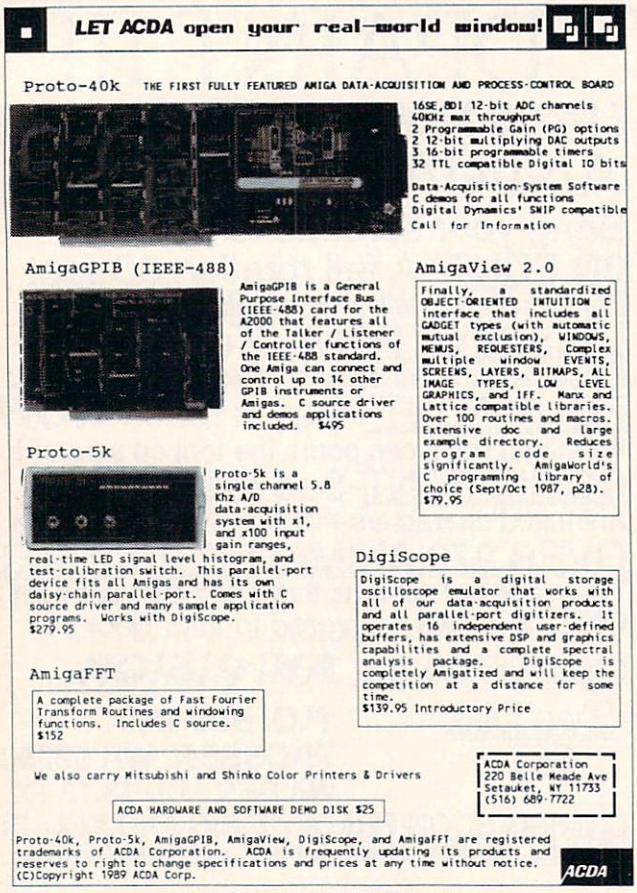
inbuf& = mybuf&
cbuf& = mybuf& + 120
ctab& = mybuf& + 240


REM - Should read  FORMnnnnACBM
rLen& = xRead&(fHandle&,inbuf&,12)
tt$ = ""
FOR kk = 8 TO 11
   tt% = PEEK(inbuf& + kk)
   tt$ = tt$ + CHR$(tt%)
NEXT

IF tt$ <> "ACBM" THEN
   loadError$ = "Not an ACBM pic file"
   GOTO Lcleanup
END IF

REM - Read ACBM chunks

ChunkLoop:
REM - Get Chunk name/length
 rLen& = xRead&(fHandle&,inbuf&,8)
 icLen& = PEEKL(inbuf& + 4)
 tt$ = ""
 FOR kk = 0 TO 3
    tt% = PEEK(inbuf& + kk)
    tt$ = tt$ + CHR$(tt%)
 NEXT
```



Circle 104 on Reader Service card.

```
IF tt$ = "BMHD" THEN   'BitMap header
   foundBMHD = 1
   rLen& = xRead&(fHandle&,inbuf&,icLen&)
    iWidth%   = PEEKW(inbuf&)
    iHeight% = PEEKW(inbuf& + 2)
   iDepth%  = PEEK(inbuf& + 8)
   iCompr%  = PEEK(inbuf& + 10)
   scrWidth%  = PEEKW(inbuf& + 16)
   scrHeight% = PEEKW(inbuf& + 18)

    iRowBytes% = iWidth% /8
    scrRowBytes% = scrWidth% / 8
    nColors%  = 2^(iDepth%)

    REM - Enough free ram to display ?
    AvailRam& = FRE(-1)
     NeededRam&  =  ((scrWidth%/8)*scrHeight%*(iDepth%+1))+5000
    IF AvailRam& < NeededRam& THEN
       loadError$ = "Not enough free ram."
       GOTO Lcleanup
    END IF

    hires& = &H8000
    lace&  = &H4
    kk = 1
    IF foundCamg THEN
       IF (camgModes& AND hires&) THEN kk = kk+1
       IF (camgModes& AND lace&)  THEN kk = kk+2
    ELSE
       IF scrWidth% >= 640 THEN kk = kk + 1
       IF scrHeight% >= 400 THEN kk = kk + 2
    END IF
    SCREEN 2,scrWidth%,scrHeight%,iDepth%,kk
     WINDOW 2,"LoadACBM",,7,2

     REM - Get addresses of structures
     GOSUB GetScrAddrs

    REM - Black out screen
    CALL LoadRGB4&(sViewPort&,ctab&,nColors%)


ELSEIF tt$ = "CMAP" THEN   'ColorMap
    foundCMAP = 1
     rLen&  = xRead&(fHandle&,cbuf&,icLen&)

    REM - Build Color Table
    FOR kk = 0 TO nColors% - 1
        red% = PEEK(cbuf&+(kk*3))
        gre% = PEEK(cbuf&+(kk*3)+1)
        blu% = PEEK(cbuf&+(kk*3)+2)
         regTemp% = (red%*16)+(gre%)+(blu%/16)
         POKEW(ctab&+(2*kk)),regTemp%
    NEXT


ELSEIF tt$ = "CAMG" THEN 'Amiga ViewPort Modes
    foundCamg = 1
     rLen&  = xRead&(fHandle&,inbuf&,icLen&)
     camgModes&  = PEEKL(inbuf&)


ELSEIF tt$ = "CCRT" THEN 'Graphicraft color cycle info
    foundCCRT = 1
     rLen&  = xRead&(fHandle&,inbuf&,icLen&)
     ccrtDir%     = PEEKW(inbuf&)
     ccrtStart%   = PEEK(inbuf& + 2)
     ccrtEnd%     = PEEK(inbuf& + 3)
     ccrtSecs&    = PEEKL(inbuf& + 4)
     ccrtMics&    = PEEKL(inbuf& + 8)


ELSEIF tt$ = "ABIT" THEN   'Contiguous BitMap
    foundABIT = 1

    REM - This only handles full size BitMaps, not brushes
    REM - Very fast - reads in entire BitPlanes
    plSize& = (scrWidth%/8) * scrHeight%
    FOR pp = 0 TO iDepth% -1
        rLen&  = xRead&(fHandle&,bPlane&(pp),plSize&)
    NEXT


ELSE
    REM - Reading unknown chunk
    FOR kk = 1 TO icLen&
        rLen&  = xRead&(fHandle&,inbuf&,1)
    NEXT
    REM - If odd length, read 1 more byte
    IF (icLen& OR 1) = icLen& THEN
        rLen&  = xRead&(fHandle&,inbuf&,1)
    END IF

END IF


REM - Done if got all chunks
IF foundBMHD AND foundCMAP AND foundABIT THEN
    GOTO GoodLoad
END IF
```

```
REM - Good read, get next chunk
IF rLen& > 0 THEN GOTO ChunkLoop

IF rLen& < 0 THEN   'Read error
    loadError$ = "Read error"
   GOTO Lcleanup
END IF

REM - rLen& = 0 means EOF
IF (foundBMHD=0) OR (foundABIT=0) OR (foundCMAP=0) THEN
    loadError$ = "Needed ILBM chunks not found"
    GOTO Lcleanup
END IF


GoodLoad:
loadError$ =""

REM  Load proper Colors
IF foundCMAP THEN
     CALL LoadRGB4&(sViewPort&,ctab&,nColors%)
END IF

Lcleanup:
IF fHandle& <> 0 THEN CALL xClose&(fHandle&)
IF mybuf& <> 0 THEN CALL FreeMem&(mybuf&,mybufsize&)

RETURN


GetScrAddrs:
REM - Get addresses of screen structures
    sWindow&    = WINDOW(7)
    sScreen&    = PEEKL(sWindow& + 46)
    sViewPort&  = sScreen& + 44
    sRastPort&  = sScreen& + 84
    sColorMap&  = PEEKL(sViewPort& + 4)
    colorTab&   = PEEKL(sColorMap& + 4)
    sBitMap&    = PEEKL(sRastPort& + 4)

    REM - Get screen parameters
    scrWidth%   = PEEKW(sScreen& + 12)
    scrHeight%  = PEEKW(sScreen& + 14)
    scrDepth%   = PEEK(sBitMap& + 5)
    nColors%    = 2^scrDepth%

    REM - Get addresses of Bit Planes
    FOR kk = 0 TO scrDepth% - 1
        bPlane&(kk) = PEEKL(sBitMap&+8+(kk*4))
    NEXT
RETURN


blowingup:
PALETTE 1,1,1,1
PALETTE 3,1,0,0

DIM p%(850,5)

PRINT "Click on upper-left of Cell 1"
WHILE MOUSE(0)<>0
WEND                          'Clear the mouse buffer
WHILE MOUSE(0)=0
  x1=MOUSE(1):y1=MOUSE(2)
WEND

PRINT "Click on lower-right of Cell 1"
```

OOPS! Corrections!

In "Fast PixQ", AC V4.11, page 19, we made several errors in the article and listing. The corrections are:
-Figure 1 and Figure 2 are reversed.
-In Figure 2, under the 'Color=5' should be '1010'.
-Paragraph 2, Column 3, page 21: "Next, isolate the individual bit" should be "individual pixel".
-Paragraph 1, Column 3, page 20: The 'AE's should be '->'s.
-Figure 2: All the '8's should be '0's.
-Paragraph 1, Column 2, page 21: The sentence "row y now has to be stored in register A2" should read, "row y now is stored in register A2." Sentence 3 should read, "It is then extended to a long integer, as done above."
-Paragraph 3, column 1, page 22: The sentence, "In the second, the byte is bset to 0" should read, "the bit is bclred to 0."
-The authors of the *Amiga Hardware Reference Manual* are Peck R., Deyl S., and Raymond C.

In "New Products and Other Neat Stuff", AC V4.11, pp. 63-64, an incorrect phone number for MichTron was printed. The correct number is (313) 334-5700. The correct price of VIVA is $199.95. Also, as no CD-ROM capability is currently available for the Amiga, VIVA is not CD-ROM supportive. Finally, the animation capabilities mentioned will be available in a forthcoming Professional version of VIVA.

In "View From The Inside", AC V4.12, page 16, an incorrect phone number for Blue Ribbon Bakery was printed. The correct number is (404) 377-1514.

We apologize for these errors and any difficulties or inconveniences that may have been caused as a result.

```
WHILE MOUSE(0)<>0
WEND
WHILE  MOUSE(0)=0
  x2=MOUSE(1):y2=MOUSE(2)
WEND

cellwidth=x2-x1

FOR x=x1 TO x1+(cellwidth*5) STEP cellwidth-1
GET   (x+1,y1)-(x+cellwidth-2,y2-1),p%(0,c)
c=c+1
NEXT

CLS

FOR l=1 TO 5
  FOR x=0 TO 5
     PUT (100,100),p%(0,x),PSET
      FOR d=1 TO 500:NEXT
  NEXT

FOR d=1 TO 750:NEXT d

  FOR x=5 TO 0 STEP -1
     PUT (100,100),p%(0,x),PSET
      FOR d=1 TO 500:NEXT
  NEXT
NEXT

WINDOW CLOSE 2
SCREEN CLOSE 2
```

•AC•

ANIM-AIDS

Where to Find the Help You Need

by Barry Solomon
Video Editor

Sometimes in the crush to find the "perfect" animation program or programs, you know, the one that really suits our own personal animating character, we can forget about the "little" things. There are a number of items available today that can help you learn new techniques, polish your style, or just make doing what you do faster or easier.

There are a number of tutorials available that can be very helpful for current or would-be animators. Available on videotape, Professional Techniques for Deluxe Paint III can be very useful, even for experienced DPaint users. Developed by Jeff Bruette's Prism Computer Graphics this 60-minute videotape features Jeff (of Max Headroom and Amazing Stories fame) giving away many of his best secrets. Almost half of the tape is devoted to such animation topics as "Animpainting", using "Animated Brushes" and use of DPaint III's Move Requester. The very down-to-earth tutorials are sprinkled throughout with thoughts from Dan Silva, author of the DPaint series. This tape really does have something for everyone, from the inexperienced to the very experienced animator.

Also in the videotape department is a series of graphics tutorials from Cape Fear Teleproductions. Micro Digital Graphics is the name of the series and Volume Two: Color Cycling Animation provides about 45 minutes of instruction on the art (and it is an art) of animating by means of color cycling. Once again, the valuable information contained in this tape is simple and well explained enough for beginners but will remind experienced animators that, sometimes, the effects they wish to achieve are easier to do than they think. Actually, I was a little chagrined by this tape. I now realize that I have become a little jaded by all the "fablulous" animation techniques available to me. I really had forgotten a lot of the really nice things you can do, very easily and quickly, with color cycling.

Last (in the video tutorials), but not least, there are a number of people out there who are still very confused and/or intimidated by The Director. This incredibly powerful program is run with a basic-like programming language. For those of you who, like me, aren't programmers, there is an answer! The Director Tutorial is a VHS videotape available to answer just about any question you may have. This 260-minute tape (that's right-four hours and twenty minutes) can answer just about any questions you have.

There are also a number of animation tutorials available on disk. RGB Computer & Video Creations, the "DeluxeHelp" people, have a set entitled DeluxeHelp for DeluxePaint III. The second disk of this two-disk set is devoted entirely to animation and provides information on a number of animation techniques. The folks at For Art's Sake also have a two-disk set that provides information on animation along with basic Amiga graphic techniques.



Just some of the many "Anim-Aids" available.

In the book department, we have the 2nd Edition of Using DeluxePaint. Written by Steve Anzovin and published by Compute! Books, I found this otherwise excellent book just a bit of a pain because it discusses DPaint for all computers (did you know that there are other computers besides the Amiga?) However, the last section is strictly about DeluxePaint III (which, of course, is

available only for the Amiga) and discusses its animation methods at length. There is even a book entitled Timing for Animation published by Focal Press. Written by two professional animators, the book deals mainly with timing techniques for film, but much of the advice, information, and hints can be applied to Amiga animation. If you are a programmer, there are any number of books available which discuss graphics and animation programming in various languages.

As for actual animation helpers, there are also a number of items available. Charles Voner of CV Designs produces the Video Visions series of disks. These multi-disk sets are designed primarily as aids for video production, but they are also helpful to anyone doing animation. The various sets and "Mini-Volumes" contain backgrounds, animated characters, animated intros, borders, logos and colorfonts which can be very useful. In particular Video Visions sets 2, 3, and 6 are particularly targeted towards animators and are available packaged together as the Animator Set.

On the humorous side, we have a two-disk set of "animated fonts" from Anivisions. While these technically are not fonts as we usually think of them, these are very interesting. Actually these are "Anim Brushes" for use in DPaint III. There are several animated fonts on these disks which are actually sets of animated brushes which must be stamped out, or "AnimPainted" in DPaint III. My favorite is the Pourfont, which produces letters, words, or entire title pages with each letter being "poured" out of a little rainbow-colored (much like the original DPaint "paint can"). Also included are a 3D (-like) font, Comic Font, and Dissolve Font. I found all of these very interesting.

Also out from The Picturebox is Build 'Em. This is a figure construction set containing over 500 anti-aliased body parts designed to be cut, rotated, and pasted together to produce the people of your choice. These parts are contained on 25 screens and were drawn by a professional figure artist.

These are only a few of the many, many ways to get more out of your adventures in animation. If I've missed one of your favorites don't be angry. There are way too many "ANIM-AIDS" to cover all at once. If you have a hint on a program, book, tutorial, or other piece which you have found to be helpful in you animating please write to me and let me know.

•AC•

Products Mentioned

Professional Techniques for DeluxePaint III
Prism Graphics
Distributed by Electronic Arts
1829 Gateway Drive
San Mateo, CA 94404
(800) 245-4525 in CA (415) 571-7171
Price: $24.95
Inquiry # 213

Micro DIgital Graphics Volume Two: Color Cycling Animation
Cape Fear Teleproductions, Inc.
605 Dick St., Wilmington, NC 28401
(919) 762-8028
Price: $39.95
Inquiry # 214

The Director Tutorial
The Right Answers Group
P.O. Box 3699
Torrance, CA 90510
(213) 325-1311
Price: $39.95
Inquiry # 215

DeluxeHelp for DeluxePaint III
RGB Video Creations
2574 PGA Blvd., Suite 104
Palm Beach Gardens, FL 33410
(305) 622-0138
Price: $49.95
Inquiry # 216

Using DeluxePaint 2nd Edition
COMPUTE! Books
324 West Wendover Ave., Suite 200
Greensboro, NC 27408
(919) 275-9808
Price: $19.95
Inquiry # 217

Timing For Animation
Focal Press, Inc.
80 Montvale Ave.
Stoneham, MA 02180
(800) 544-1013
Price: $37.95
Inquiry # 218

Video Visions: Various DIsks
CV Designs
61 Clewly Rd.
Medford, MA 02155
(617) 396-8354
Prices Vary With Sets
Inquiry # 219

Animated Fonts
Anivision
P.O. Box 801
Prosser, WA 99350
(509) 973-2717
Price: $39.95
Inquiry # 220

Build 'Em
The Picture Box
8824 David Ave.
St. John, MO 63114
(314) 427-1869
Price: $23.50
Inquiry # 221

DUAL DEMO

Programming an Arcade Game Can Be Easy

by Thomas J. Eshelman

If you type in and study the code that accompanies this article, you will immediately see how it provides the foundation for neat, professional looking games limited in scope only by your own imagination. The secret lies in using the Amiga's dual playfields. One contains and scrolls objects drawn with your favorite paint program, while the other can either be similarly scrolled or remain stationary. Scrolling a few objects gives the appearance of many individually "animated" objects (or bobs), making the program appear far more complex than it is.

"DualDemo" appears as follows: You look at a yellow background through a window outlined in red. White text—including "score," which changes as the program progresses—appears on the top and bottom borders of the red window. "Animated" colored squares appear against the yellow background, rising majestically from bottom to top. With the joystick, you control an electric-green sprite "spaceship." Whenever the spaceship hits a red square, the score increments by a value of five, whereupon the program exits. A black band remains on either side of the scrolling playfield(pf) to prevent the optical illusion of looking through a window—the illusion you usually desire when writing games.

Once you see how this is done, you need only vary the sizes, colors and shapes of the objects, adding any audio/visual collision routines you desire. By adding appropriate scorekeeping routines, you can design your own game-of-the-week.

The code is relatively simple. The only difficulties lie in interpreting the instructions given in the ROM Kernel Manual, and in a shortage of published source codes after which you can model your own. Due to the space limitations of this article, I will discuss only those techniques not otherwise available to the average reader.

I would first like to give you some good advice based on my own experience: learn assembly! The only connection I have with computers is my Amiga hobby. I fooled with BASIC and C for nearly two years, and by memorizing many procedure calls, I was able to create decent programs. But, whenever something went wrong... aargh!

Only via assembly can you fully comprehend what's happening when high-level languages execute, and only then can you take corrective measures based on your own logic. Header files will become comprehensible! The beauty of it all is that assembly is by far the easiest of the languages for a rank beginner to learn. There are many good and inexpensive assemblers. The ugly part is a lack of good tutorial texts.

If you have a college or university in your area, I would suggest at least auditing an assembler course. I took a formal course in VAX assembler last spring, and found it invaluable. VAX and Amiga assembler are extremely similar. I alternate time between assembly and C, just to retain a familiarity with both. I unequivocally recommend assembly as your first programming language.

DUAL VS. SINGLE PLAYFIELDS

Dual playfields require the employment of two bitmaps, two rasinfos, and two rastports. It is also advisable to create and swap "Copper list" pointers, not only to create truly exotic dual playfields, but also to serve as an introduction to double-buffering.

IN GENERAL

Displays and objects are drawn into Bitmaps, which are composed of planes. The corresponding bits of the planes being set—or not set—indirectly determine the color for that bit (pixel) location within the overall Bitmap. This is because these bits reveal, in binary, the color register number which will be displayed there. The user-defined color table reveals the actual colors themselves,



beginning with register 0 and running as high as register 31. The Bitmap may be much larger than the display seen by the user.

The skeleton of any display is called the View. It can be fleshed with more than one ViewPort, but usually it is not. In graphics, the ViewPort is the CustomScreen in Intuition. The Layers of graphics are the Windows of Intuition. For our purposes, the display is the ViewPort, since we only have one.

A RastPort is a skeleton whose bones consist of all the drawing parameters, so all drawing routines look to the RastPort to interpret the arguments you give them. Each ViewPort or Layer must have its own RastPort if you expect to do any drawing. The flesh of the RastPort is the Bitmap into which graphics are rendered.

The purpose of this article is to make clear certain unorthodox, but immutable, rules that govern all the above procedures related to dual playfields and collision detection. They are really quite simple, although difficult to explain. Lets summarize some of them.

SPECIAL RULES FOR DUAL PLAYFIELDS

RULE 1: The first Bitmap initialized becomes playfield 1 (pf1). pf1 automatically gets priority over (appears in front of) pf2, unless the PFBA bit is set in ViewPort.Modes.

RULE 2: pf1 cannot have fewer planes in its Bitmap than pf2 has in its Bitmap. If you allocate five out of the permitted six bitplanes for our program, three will go to pf1 and two will go to pf2. Therefore, you would have eight colors available in pf1 and four in pf2. To see more colors in the background than in the foreground, change their relative priority with the PFBA bit as illustrated in Table One.

RULE 3: The planes of pf1 are system-assigned to odd numbers (1, 3 and 5). The evens (2 and 4) go to pf2. In the language of Commodore, the bitplanes are numbered from 1 to 6 inclusive. The bits, however, are numbered from 0 to 15. Watch out, the first color of each pf is not a color at all; it is "transparent." These facts are inherent in the Amiga, so don't try to fudge it.

RULE 4: The color registers used for display in the dual-pf mode are fixed (see Table Two). In this example, registers 0 through 7 are used for pf1, while 8 through 11 are used for pf2. Confusion may result when setting pens and the like in pf2 (i.e., the eighth register holds color 0 for pf2).

Kindly look at the dimensions of the playfields in the code. pf1 is a multiple of 320 + 1. My 960 + 1 is something of a waste, but I put it there to encourage experimentation. When the program starts, 320 lines appear in the backfield. As these scroll upward, 320 more lines must be available to replace them. When these are all fully displayed, you immediately redraw from the beginning by resetting your RasInfo.RyOffset to 0. Therefore, pf1 (which has more colors) really works just as well at 640 + 1. pf2 is set at the typical 320 x 200, since it remains centered in the ViewPort. You won't crash the machine if you fail to add that one extra line to the Bitmap, but you won't like the display appearance as much.

While we are on the subject of colors, let us prepare for the collision-detection scheme. Observe that the red square in pf1 appears in color register 4 (0xd000). Remember what I said earlier: 4 in binary (0100) would seem to mean that wherever you have a bit set in plane 3, and in no other, a red pixel will appear on your display. Well, you're right—only it's not the third plane, but the fifth. Gotcha! Remember, pf1 can only have the odd planes, 1, 3 and 5. Hence, the third is really the fifth.

Table One

No. of Bit Planes Allocated	pf1 value	pf2 value
0	0	0
1	1	0
2	1	1
3	2	1
4	2	2
5	3	2
6	3	3

Table Two

Depth, pf 1	System Registers	Depth, pf 2	System Registers
1	0-1	1	8-9
2	0-3	1	8-9
2	0-3	2	8-11
3	0-7	2	8-11
3	0-7	3	8-15

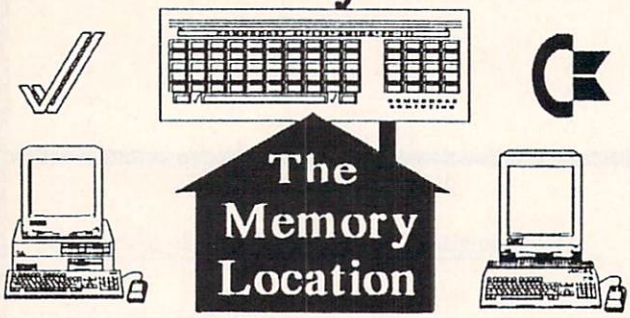
Table Three A: The clxdat register (read only)

Bit No.	What it reports
0	Even bitplanes to dd bitplanes.
1	Odd bitplanes to sprite 0.
2	Odd bitplanes to sprite 2.
3	Odd bitplanes to sprite 4.
4	Odd bitplanes to sprite 6.
5	Even bitplanes to sprite 0.
6	Even bitplanes to sprite 2.
7	Even bitplanes to sprite 4.
8	Even bitplanes to sprite 6.
9	Sprite 0 to sprite 2.
10	Sprite 0 to sprite 4.
11	Sprite 0 to sprite 6.
12	Sprite 2 to sprite 4.
13	Sprite 2 to sprite 6.
14	Sprite 4 to sprite 6.
15	Unused.

Table Three B: The clxcon register (write only)

Bit No.	What it does
0	Enable bitplane1 collisions.
1	Enable bitplane 2 collisions.
2	Enable bitplane 3 collisions.
3	Bitplane 4 collisions.
4	Enable bitplane 5 collisions.
5	Enable bitplane 6 collisions.
6	Validate or bias bitplane 1.
7	Validate or bias bitplane 2.
8	Validate or bias bitplane 3.
9	Validate or bias bitplane 4.
10	Validate or bias bitplane 5.
11	Validate or bias bitplane 6.
12	Enable sprite 1 (detects if sprite 0 collides).
13	Enable sprite 3 (detects if sprite 2 collides).
14	Enable sprite 5 (detects if sprite 4 collides).
15	Enable sprite 7 (detects if sprite 6 collides).

Come see whats Hot
for the *AMIGA* at
The Memory Location



396 Washington Street
Wellesley, MA 02181
(617) 237-6846

AMIGA Experts! Nothing but the best.
Satisfaction guaranteed.
Come in and try-out all the latest software,
hardware, and accessories.

Your full service *AMIGA* dealer.

Store hours: Mon.-Thur. 10-6 Friday 10-8 Saturday 9-5.
Full Commodore authorized repair service onsite.
Low flat rate plus parts.

Circle 107 on Reader Service card.

You must understand this in order to perform the seemingly simple task of setting and interpreting the two registers. They will cause collision detections between a given sprite and a playfield object of a given color, or between two playfield objects of different colors. To talk in colors, you must talk in planes while looking at your color table.

COLLISION DETECTIONS

Take a look at Tables 3a and 3b. They contain the results of setting each of the possible 16 bits in the write-only register known as "clxcon," and of reading each of the possible bits in the one-shot, read-only register known as "clxdat." Each of these are offsets (members) of the base of the structure known as "Custom."

To read the clxdat register, you perform a logical AND on it with a mask of your own making. If the result of the AND is TRUE, you get a collision. Each move requires a separate read, because the act of reading resets the bits of clxdat. You have only one chance, unless you copy the results first.

What do you need to see in clxdat in order to register a collision of the sprite with a red block? Observe that the sprite selected in our code is No. 2. As you can see from the color table, the red rectangle in pf1 (0xd000) appears in register No. 4 (counting the first as No. 0). Binary 4 is 0100. Even though the bit is in position 2, it lies in plane 3, because the Amiga counts bitplanes from the right, beginning with a value of 1.

In the language of dual playfields, you will recall from what has been discussed above that the third bitplane of pf1 is really bitplane No. 5 of the Amiga (grin). Therefore, you want to collide sprite No. 2 with an odd-valued bitplane (No. 5). As you see from the tables, bit 3 of the clxdat register will be set by the system in the event that sprite No. 2 collides with anything in an odd-valued bitplane. Accordingly, our mask must also set the third bit, 0100 binary (or 0x004), so that the result of a logical AND is TRUE.

The clxcon register is the one to which you WRITE. It is used to narrow the field. You don't want a reaction to all odd planes that are contacted by sprite No. 2; you want a reaction only to plane 5. To do this, you need to validate or "bias" any and all bits that could be involved in the scheme, whether you want them set or not. It's like tenpins: You must set 'em up in order to knock 'em down. That means you must bias all odd planes. You want planes 1 and 3 to be "hot" so they recognize that you are notifying them not to set, just as you want 5 to be "hot" so you can tell it you do want it set.

Set the "bias" bits, which—as you can see from the table—will be numbers 6, 8 and 10. Now they will all look to their corresponding signal or command bits, which are 0, 2 and 4. Thus, you will set bits 0 and 2 to 0, which will specifically disable collision detection in planes 1 and 3. You will set bit 4 to 1, enabling detection in plane 5.

In binary, the appearance of clxcon will thus be: 0000 0101 0101 0000. In other words, the fourth, sixth, eighth and tenth bits (or hex 0x0550) are being set. This is the reason the code contains the expression:

```
cstm -> clxcon = 0x0550;
```

RELEASING THE GAME PORT

Now that I have your undivided attention, I will reveal a little known (or at least little published) secret about how you can divest the joystick of control over the computer after a program ends and the CLI prompt is returned. You often come across programs that seem to respond miraculously to stick or fire-button activity after having terminated. (I've written a few like that myself.) The secret lies in resetting the controller type to GPCT_NOCONTROLLER prior to exiting the code. To facilitate your doing this, DualDemo.c features a short routine, set_controller_type(). Put it into your library of C routines.

DOUBLE-BUFFERING THE COPPER LISTS

Finally, let's take a moment to examine and study "Copper lists." The Copper may be looked upon as the Amiga's special-hardware-chips-register-loader. When you call the routine, MrgCop(), the system scans through all the data structures you have set up and creates "lists" of assembly-like instructions for the Copper to execute. This code gives the Copper the user-input values to load in all the various hardware registers (the ones referred to in the struct Custom), and tells it when to do it. Read on carefully...

To do this, the system dynamically allocates memory for a "long-frame" Copper list (LOFCprList) and for a "short-frame" Copper list (SHFCprList). The latter is used only if you ask for an interlaced display. For the purpose of linking these lists into the overall "skeleton," the system sets pointers to these lists, in the memory set aside for them in struct View. Examine include/graphics/view.h.

The lists refer to the underlying Bitmap. With dual pf's or with double-buffering, you have provided two Bitmaps. Now what? You fool the system into creating and storing two sets of CprLists. Remember that these lists are telling the Copper things like which colors will be jammed into place for the display to which it refers. Therefore, you must acquire the habit of providing one such CprList for each Bitmap.

Let me spell it out a little more clearly:

1. The first of the "CprLists," prepared by the system in all events, will refer to the first of the two Bitmaps in dual pf's or double-buffs.

2. Therefore, after you call MrgCop() for the first time:

a. Copy View.SHFCprList and View.LOFCprList to a pair of extra pointer variables you named and allocated ahead of main()

b. Assign the pointer to your second Bitmap to the Bitmap member of your only RasInfo (in the case of double-buffs) or to that of your alternate RasInfo (in the case of dual-pf's).

c. Set the values of both "CprList" members of struct View to zero.

d. Call MrgCop() again. The system now thinks that it forgot to prepare an area for copper lists, and will generate a second pair of lists plus a second pair of pointers thereto.

e. Save the second set of pointers as you did the first. Voilà! You have two sets of instructions somewhere in memory and two sets of pointers to them, one for each of your two Bitmaps.

3. Thereafter, you swap pointers as you swap Bitmaps. Assign the proper pointer values to their proper positions in struct View prior to displaying that Bitmap, but after having copied the pair of pointers that were already in there for temporary safekeeping. Thus, each of the Bitmaps will have its proper attributes displayed along with it.

4. When you free memory at the end of your program, call FreeCprList() twice, since you called MrgCop() twice.

That's pretty much the story. Assuming you study and understand the readily available material on how to produce and draw into simple displays on the Amiga, DualDemo should make clear those parts of the RKM which, although somewhat turgid, are necessary to producing quality arcade games. It really is not all that difficult. Analyze the source code carefully, and by all means experiment with all the constant values, one or two at a time. You'll be writing—and hopefully uploading—some nice material for all of us to enjoy in no time at all.

Note: The system is never able to determine which member of a sprite pair was involved in a collision. You may enable the odd sprites, but apliprog must be capable of determining which of the pair was involved when a collision is reported. I examine sprite co-ordinates to make this determination.–N.B.

Listing One

```
/*DualDemo-by Thomas J. Eshelman - Reading, Pa.- 9/13/88
This program demonstrates dual playfield and collision detection code
for the Amiga. A yellow field appears behind a red foreground
containing text. This backfield contains an array of variously colored
squares which scroll upward. A 'spaceship' sprite is given the user
which can be controlled horizontally with the joystick. Whenever the
sprite collides with a red square, the score is incremented on the
front playfield. When the score becomes '5', the program quits.*/


#include <exec/types.h>
#include <intuition/intuition.h>
#include <graphics/gfxbase.h>
#include <graphics/sprite.h>
#include <devices/gameport.h>
#include <hardware/custom.h>
#include <functions.h>

#define DEPTH1   3L          /*pf1 = taller than display*/
```


The basics. Space and time. From an ice-age antler minutely etched with lunar phases to the inconceivably detailed information patterns transmitted by the Hubble Space Telescope, man has sought answers to basic questions about the universe in astronomy.



The best. Turn off the lights. Distant Suns, home grown on the Amiga™, presents the subtle colors of the night sky. No other astronomy program on ANY platform can do that.

Get it. A real time machine. A toy universe for your Amiga™ – Distant Suns. Virtual Reality Laboratories, Inc. (805) 545-8515.

Circle 121 on Reader Service card.


```
#define WIDTH1   230L          /*a/k/a  BitMap 1*/
#define HEIGHT1  961L

/* Note:  Since the squares pattern burns 320 lines.  Seamless scrolling
requires 640 lines.  As one pattern goes o.o.s. up, the next pattern
comes into view.  When 320 lines have scrolled, reset RyOffset to 0.The
960 * 640 setup is not required, but is utilized for illustration
only.*/

#define DEPTH2   2L      /* pf2 a/k/a BitMap 2.   */
#define WIDTH2   320L         /* Same size as display. */
#define HEIGHT2  200L

#define VPWIDTH  320L         /* ViewPort dimensions.     */
#define VPHEIGHT 200L         /* Same as overall display. */

#define MAXSCROLL_X  0L
#define MAXSCROLL_Y  640L/* HEIGHT1-HEIGHT2 = the maximum */
                              /* 'legal' scroll distance.      */
#define SAUCER_WIDTH   16
#define SAUCER_HEIGHT  13

struct View     v;
struct ViewPort vp;
struct ColorMap *cm;     /* GetColorMap() sets this up. */

struct BitMap   b1,  b2;
struct RastPort rp1, rp2;
struct RasInfo  ri1, ri2;

LONG  i, x, y, loop;
UWORD colmask;

char  score[] = { '0', '1', '2', '3', '4', '5' };

struct Custom  *cstm = (struct Custom*) 0xdff000;
                              /* Hardware rgstr base */
struct GfxBase *GfxBase;
struct View *oldview;         /* Save old view for restoration */

struct InputEvent  joyreport;
struct IOStdReq   *gameio;
struct MsgPort    *gameport;


struct GamePortTrigger gpt =  /*Reportable input conditions. */
{
```

```
    GPTF_UPKEYS | GPTF_DOWNKEYS,
    0,
    1, 1
};


UWORD colortable[] =
{                       /* First color = display background */
    0x000, 0xf70, 0x070,0x007, /* transparent, yellow, green,
                                  blue */
    0xd00, 0xfff, 0x704, 0x077,/* red, white, violet, aqua */
                               /* Below are for pf2.*/
    0x000, 0xf00, 0xfff, 0x00f /* transparent, red, white,
                                  blue */
};


UWORD  *colorpalette;


                        /*  Pairs of Copper List pointers. */
                        /*  See article.                   */
struct cprlist *LOF[2];
struct cprlist *SHF[2];
long w;


USHORT Saucer_data[]=  { /* Sprite shape definition data. */

     0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
     0x0000,0x0000,0x03c0,0x0000,0x07e0,0x0000,0x1ff8,0x0000,
     0xffff,0x0000,0xea57,0x15a8,0xffff,0x0000,0x1ff8,0x0000,
     0x07e0,0x0000,0x03c0,0x0000,0x0000,0x0000
};


struct SimpleSprite  Saucer = {

    &Saucer_data[0], SAUCER_HEIGHT,
    160, 100, 0
};



void  main()
{
    openstuff ();

    make_playfields();

    init_joystick();

    paint_scene();

    go_to_it();           /* Action! */

    Close_All();
}



init_joystick ()
{

    (void) set_controller_type ( (char) GPCT_ABSJOYSTICK );

      gameio -> io_Command = GPD_SETTRIGGER;
      gameio -> io_Length = (long) sizeof (gpt);
      gameio -> io_Data = (APTR) &gpt;

    if (DoIO (gameio))

       die ("Error in setting trigger values.\n");

    gameio -> io_Command = GPD_READEVENT;
    gameio -> io_Length = (long) sizeof (joyreport);
    gameio -> io_Data = (APTR)&joyreport;
    gameio -> io_Flags = 0;
}



go_to_it()
{
    short  hit_ok;
    short  j;

    cstm->color[17] = 0xf00;    /* Hide Intuition pointer.*/
    cstm->color[18] = 0xf00;
    cstm->color[19] = 0xf00;
    cstm->color[21] = 0x0f0;    /* Render the saucer green.*/


    cstm->clxcon =  0x0550;     /* See article for details.*/
    colmask = 0x0004;      /* Enable odd planes to spr 2.  */
    w = 0L;

    LOF[1] = NULL;              /* Because these begin nulled,   */
    SHF[1] = NULL;              /* first use creates a new list. */

    SendIO (gameio);

    hit_ok = 0;
    j = 1;

    while (1)
    {
       Saucer.x += joyreport.ie_X;

       if ( Saucer.x < 44 )              /* Sprite position is   */
          Saucer.x = 44;            /* relative to ViewPort */

       if ( Saucer.x > 259 )
          Saucer.x = 259;

       MoveSprite ( &vp, &Saucer, (long) Saucer.x, 100L );


       if ( colmask & cstm-> clxdat ) /*Check for a collision.*/
       {                              /* The hit_ok algorithm to allow  */
          if ( hit_ok )               /* only one 'score' per red hit.  */
          {                 /* (Else would keep 'scoring' til */
             hit_ok = 0;  /* sprite breaks into the clear.  */
             show_score(j);
             j++;
          }

          else
          {
             hit_ok = 0;
          }
       }

       else
       {
          hit_ok = 1;
       }

       if ( j == 6 )
       {                              /* Clear your reply port.*/
          GetMsg (gameport);    /* Then disengage joystick */

          (void) set_controller_type ( (char) GPCT_NOCONTROLLER );

          Delay (50L);                /* System may crash if it  */
          break;                /* returns to CLI too fast */
       }

       swapPointers();

       ril.RyOffset++;              /* 'Up' */

       remakeView();

       w ^= 1L;                 /*EOR causes pointers to  */
                                /*swap between [0] and [1]*/

       if ( ril.RyOffset >= MAXSCROLL_Y )

          ril.RyOffset = 0;         /*SCHNELL! Back to the top!*/

       gameio -> io_Command = GPD_READEVENT;
       gameio -> io_Length = (long) sizeof (joyreport);
       gameio -> io_Data = (APTR)&joyreport;

       GetMsg (gameport);       /* Take message off port. */
       SendIO (gameio);
    }                           /* forever loop */
}                               /* end go_to_it */



swapPointers()           /*Double-buffer the copper lists.*/
{
   LOF[w] = v.LOFCprList;/* Alternately save to [0] or [1]*/
   SHF[w] = v.SHFCprList;
                            /* And replace with EORd version.*/
   v.LOFCprList = LOF [(w^1)];
                            /* Two pairs - use same View.*/
   v.SHFCprList = SHF [(w^1)];
}


remakeView()
{
   MakeVPort(&v, &vp);      /* Use that alternate CprList */
   MrgCop(&v);
   LoadView(&v);            /* Show new display. */
   WaitTOF();               /* Slow it down.     */
```

```
}


die (str)                   /* A little Schwabism.*/
char *str;
{
   puts (str);
   Close_All ();
   exit (100);
}


openstuff()
{
   if (!(gameport = CreatePort (0L, 0L)))

      die ("Can't make gameport.\n");

   if (!(gameio = CreateStdIO (gameport)))

      die ("Can't make game IOReq block.\n");

   if ((OpenDevice ("gameport.device", 1L, gameio, 0L)) != NULL )

      die ("Can't open the game device.\n");


   GfxBase = (struct GfxBase *)OpenLibrary("graphics.library",0L);

   if (GfxBase == NULL)
      die ("Can't open graphics.library.\n");

  /* Save the View of whatever you ran this program from. */

   oldview = GfxBase->ActiView;

   return;
}


paint_scene()
{
   short  spr;

   SetAPen(&rp2,1L);      /*Cover pf2 with a red rectangle.*/
   RectFill(&rp2,0L,0L,320L,200L);

   SetAPen(&rp2,0L);      /* OPEN 240 HOLE to see backfield.*/
   RectFill(&rp2,40L,40L,280L,160L);
                          /* 40 wide margins all around.*/

   SetAPen(&rp1,1L);      /*Draw yellow rect all over pf1.*/
   RectFill(&rp1,0L,0L, 230L,960L);  /* Offsets from Dx/Dy.*/

   y = 0L;           /* Draw colored squares all over.*/

   SetAPen ( &rp1, 6L);

   for ( loop = 0; loop <= 2; loop++ )       /* 3 draws */
   {
      for ( i = 0, x = 25; i <= 4; i++, x += 40, y += 40 )
                                             /* 5 right */
      {
         RectFill ( &rp1, x, y, x + 20L, y + 20L );

         SetAPen (&rp1, i + 2L );
      }

      for ( i = 0, x = 145; i <= 2; i++, x -= 40, y += 40 )
                               /* 3 left */
      {
         RectFill ( &rp1, x, y, x + 20L, y + 20L );

         SetAPen ( &rp1, i + 3L ); /*Each 8 takes 320 lines*/
      }
   }

   SetDrMd ( &rp2, JAM2 );          /*Text is measured from top.*/
   SetAPen ( &rp2, 2L );       /* Chars are 8 pixels wide.*/
   SetBPen ( &rp2, 1L );

   Move ( &rp2, 80L, 14L );
   Text ( &rp2, "Dual Playfields Demo", 20L );

   Move ( &rp2, 100L, 30L );
   Text ( &rp2, "by Tom Eshelman", 15L );

   Move ( &rp2, 106L, 183L );
   Text ( &rp2, "Hits On Red:", 12L );

   Move ( &rp2, 209L, 183L );
   Text ( &rp2, &score[0], 1L );
```


We take a byte out of the price



ONE BYTE

P.O. Box 455
Quaker Hill, CT 06375
(800) 441-BYTE, in CT (203) 443-4623

Authorized dealer for
Commodore-Amiga Computers,
Great Valley Products (GVP),
Memory & Storage Technology (M.A.S.T.).
Authorized Commodore-Amiga Service and Repair.
Authorized Amiga Graphics Dealer.

AMIGA IS A REGISTERED TRADEMARK OF COMMODORE-AMIGA, INC.

Circle 135 on Reader Service card.


```
   spr = GetSprite ( &Saucer, 2L );/* Demand sprite no. 2.*/

   if ( spr != 2 )

      Close_All();

   ChangeSprite ( &vp, &Saucer, &Saucer_data[0] );

   Delay (75L);          /* Pause for the draw. */
   return;
}


make_playfields ()
{
   InitView(&v);
   v.ViewPort = &vp;

   InitVPort(&vp);
              /*First register of pf2 is No. 9. It is xparent.
            It is useful to experiment with this.*/

   cm = (struct ColorMap*) GetColorMap (12L);
   colorpalette = (UWORD *) cm->ColorTable;

   for(i = 0; i < 12; i++)
      *colorpalette++ = colortable[i];
                              /*Copy colors to data structure.*/
   vp.ColorMap = cm;      /* Link same. */

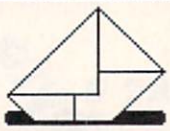
   vp.DWidth  = VPWIDTH; /* What user actually sees */
   vp.DHeight = VPHEIGHT;/* i.e:  the Display. */
   vp.RasInfo = &ri1;

   vp.Modes   = DUALPF | PFBA;/*Swap priorities so you can */
                              /* have more colors in backfield */

   LoadRGB4 ( &vp, &colortable[0], 12L );
                              /* Load color registers. */

   InitBitMap(&b1,DEPTH1,WIDTH1,HEIGHT1);
   InitBitMap(&b2,DEPTH2,WIDTH2,HEIGHT2);
```

MOVING?



SUBSCRIPTION PROBLEMS?

Please don't forget to let us know.
If you are having a problem with your subscription or if you are planning to move, please write to:

Amazing Computing Subscription Questions
PiM Publications, Inc.
P.O. Box 869
Fall River, MA 02722

Please remember, we cannot mail your magazine to you if we do not know where you are.

Please allow four to six weeks for processing.


```
for(i = 0; i < DEPTH1; i++)
{
   b1.Planes[i] = (PLANEPTR) AllocRaster (WIDTH1, HEIGHT1);

   if (b1.Planes[i] == NULL)

      die ("Can't open pf1 bitplanes.\n");

   BltClear (b1.Planes[i], RASSIZE ( WIDTH1, HEIGHT1 ), 0L);
}

for(i = 0; i < DEPTH2; i++)
{
   b2.Planes[i] = (PLANEPTR) AllocRaster ( WIDTH2, HEIGHT2);

   if (b2.Planes[i] == NULL)

      die ("Can't open pf2 bitplanes.\n");

   BltClear (b2.Planes[i], RASSIZE ( WIDTH2, HEIGHT2 ), 0L);
}

            /* How to init RasInfo structures!*/
ri1.BitMap   = &b1;
            /* ViewPort to be positioned way to the left  */
ri1.RxOffset = -45;
            /* 40 for border + 5 as study aid.  It is the */
ri1.RyOffset = 0;
            /* BITMAP you draw to.  NOT the ViewPort!*/

ri1.Next = &ri2;

ri2.BitMap = &b2;

ri2.RxOffset = 0;
ri2.RyOffset = 0;

ri2.Next = NULL;
MakeVPort( &v, &vp );       /* Construct copper instr
                            (prelim) list */

   MrgCop( &v );                    /* Merge prelim. lists into a
                                    final     */
         /* copper list in the view structure.   */

   InitRastPort(&rp1);         /* Need a pf1 rastport to draw. */
   rp1.BitMap = &b1;      /* Link it to its bitmap.        */

   InitRastPort (&rp2);
   rp2.BitMap = &b2;

   SetRast(&rp1,0L);/*  Raster bitmaps transparent. */
   SetRast(&rp2,0L);

   LoadView(&v);
}

Close_All()
{
   for(i = 0; i < DEPTH1; i++)
   {
      if (b1.Planes[i]) FreeRaster ((char*) b1.Planes[i], WIDTH1,
HEIGHT1);
   }

   for(i = 0; i < DEPTH2; i++)
   {
      if (b2.Planes[i]) FreeRaster ((char*) b2.Planes[i], WIDTH2,
HEIGHT2);
   }
         /* Free structures 'dynamically'  */
         /* created by the system for you. */
   if (cm)

      FreeColorMap(cm);

   FreeVPortCopLists (&vp);

   FreeCprList (LOF[0]);
   FreeCprList (LOF[1]);

   if (Saucer.num)

      FreeSprite ( 2L );

   if (gameio)
   {
      if (gameio -> io_Device)

         CloseDevice (gameio);

      DeleteStdIO (gameio);
   }

   if (gameport)

      DeletePort (gameport);

   LoadView(oldview);              /* Replace old view */

   Delay(50L);                  /* Be sure it's in place */

   CloseLibrary((long)GfxBase);

}


show_score(k)
short k;
{
   SetDrMd ( &rp2, JAM2 );
   SetAPen ( &rp2, 2L );
   SetBPen ( &rp2, 1L );

   Move ( &rp2, 209L, 183L );
   Text ( &rp2, &score[k], 1L );
}

/* This long-unsung technique prevents the joystick from affecting the
computer in any way after the program is done and you get your prompt
back.  For this reason, I have prepared a special routine for it, so
that its importance is not lost on the reader.*/

set_controller_type (type)
char  type;
{
         gameio -> io_Command = GPD_SETCTYPE;
         gameio -> io_Length = 1L;
         gameio -> io_Data = (APTR) &type;

         if (DoIO (gameio))
            die ("Error in setting controller type.\n");
}
```

•AC•

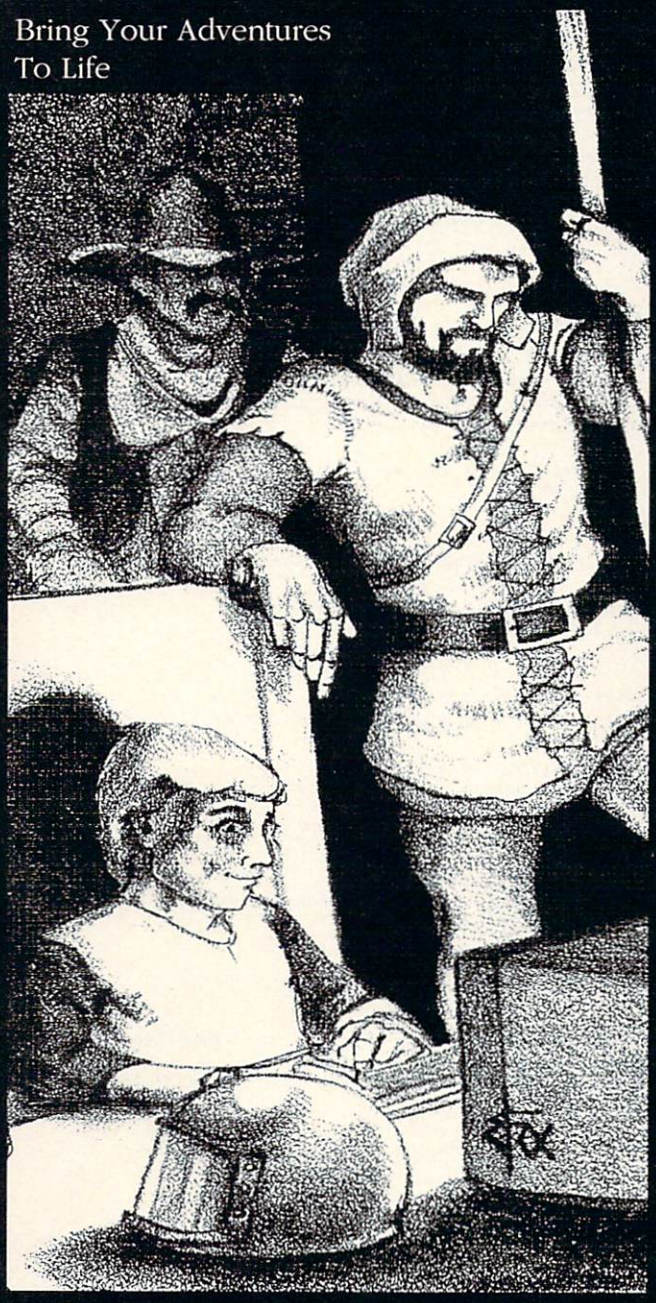
NEW RELEASE!!

T.A.C.L.

The Adventure Construction Language

Written by Alternate Realities. Produced by Micro Momentum, Inc.

Bring Your Adventures
To Life



T.A.C.L. allows you to unleash the creativity of your mind and bring it to reality. T.A.C.L. is a full-featured text/graphics adventure language.
Anything your mind can imagine can be brought to reality using T.A.C.L. Into the future, back in time, into the 5th dimension—you are in control. T.A.C.L. has the following features:

-Supports IFF graphics (including HAM), sound, different text styles, vector graphics, and more.

-Includes PADV, the adventure player that is freely-redistributable so you can distribute it with your favorite public domain adventures.

- Includes VGED, a complete vector graphics editor that allows you to have many drawings with each adventure without taking up large amounts of disk space.

- T.A.C.L. source code can be ported and compiled with no modifications on other computers that T.A.C.L. is available for (IBM and Mac versions in development).

- T.A.C.L. will allow you to create commercial-quality adventure games. Send us a copy and we will consider distributing it for you!

- Runs on all Amigas with 512K RAM and AmigaDOS V1.2 or above.

-Available at Amiga dealers or direct.

Micro Momentum, Inc.
P.O. Box 372
Washington Depot, CT 06794
Order Phone: 800-448-7421

Suggested Retail $99.95

We want to make you a success by producing your software. We deal fair and square so give us a call. Make the first move.

Circle 125 on Reader Service card.

(continued from page 26)

one menu is the combined lengths of the previous ones! For the Items, the position is lined up with the menu they belong to. sub-Items are a positioned a little differently. The Intuition manual states "The sub-Item box must overlap its item's select box somewhere". This seems clear enough except that while a 10% overlap looks kind of neat when the sub-Item pops out of the item; it is hard to hit with the mouse pointer! The problem is that an item may deselect while you are moving the mouse to select the sub-Item! If the overlap is small, a slight jiggle in the mouse may select the next menu item and make your software very touchy. To solve this problem, overlap the sub-Item half way across Item and drop it down just under half of the height of an item. This gives plenty of overlap and doesn't totally hide the items.

That completes the MenuBuilder. It is possible to carry this program a little farther by actually completing the many structures and creating a real menu that can be attached to the current window. This would eliminate the need to compile and link a menu to test it. This would require additional work, and since I am the lazy one, I'll leave that up to you!

LISTING ONE

```
/* menu test driver program  V2.0
This program will open a screen and a window just so we
can test the menu.  We could find out the current window
and use it for the menu, but this is a more common usage!
*/
#include <exec/types.h>
#include <intuition/intuition.h>
#include <graphics/display.h>
#include <libraries/dos.h>
#include <libraries/dosextens.h>
#include <stdio.h>
#include <stdlib.h>
#include <proto/intuition.h>
#include <proto/graphics.h>
#include <proto/exec.h>
#include <string.h>
struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase;
extern struct Menu *MyMenu;
struct NewScreen NewScreen =
  {
  0,0,640,200,          /* Left, Top, Width, Height */
  2,0,1,                /* Depth, detail Pen, Block Pen*/
  HIRES,                /* Hi resolution screen */
  CUSTOMSCREEN,         /* type */
  NULL,                 /* Font */
  "Custom Services"     /* Screen Title */

  };
struct NewWindow NewWindow =
  {
  0, 0, 640, 200,       /* Left edge, Top edge, width, height
*/
  0, 1,                   /* Width, Height */
  CLOSEWINDOW | MENUPICK, /* IDCMP message flags */
  WINDOWCLOSE | WINDOWSIZING | WINDOWDRAG | WINDOWDEPTH |
  SMART_REFRESH | ACTIVATE | NOCAREREFRESH,
  NULL,                    /* First Gadget */
  NULL,                    /* default to checkmark */
  "Menu Test Window",      /* Title of window      */
  NULL,                    /* Standard Screen      */
  NULL,                    /* Bit Map              */
  100,  25,                /* Min Width, Min height*/
  640, 200,                /* Max width, Height    */
  CUSTOMSCREEN             /* Screen type          */

  };
int Flgs[26];          /* one flag for each letter */
/*
** compiler prototypes
*/
void Do_Arguments(int, char **);
void Process_Item(int, int, int);
/*
** Main Program
*/
void main(argc, argv)
int argc;
char **argv;
  {
  unsigned char str[30];        /* window message structure */
  unsigned int i;               /* loop counter */
  unsigned M0, I0, S0;          /* Menu control index */
  struct Screen *Screen;        /* ptr to screen*/
  struct Window *Window;        /* ptr to window*/
  struct IntuiMessage *message; /*message pointer */
  /*
  ** ignore arguments, open libraries and startup
  */
  IntuitionBase = (struct IntuitionBase
*)OpenLibrary("intuition.library",0);
  if( IntuitionBase == NULL )
    {
    printf("Unable to Open intuition.library");
    exit(10);

    };
  GfxBase = (struct GfxBase
*)OpenLibrary("graphics.library",0);
  if( GfxBase == NULL )
    {
    printf("Unable to Open graphics.library");
    exit(10);

    };
  if( (Screen = (struct Screen *)OpenScreen(&NewScreen))==NULL)
    {
    printf("Unable to open Screen");
    exit(10);

    };
  NewWindow.Screen = Screen;
  if( (Window = (struct Window *)OpenWindow(&NewWindow))==NULL)
    {
    CloseScreen(Screen);
    printf("Unable to open window");
    exit(10);

    }
  /*
  ** Here we tell the system about our menu!
  */
  SetMenuStrip(Window, MyMenu);
  /*
  ** Display something in our window
  */
  Move(Window->RPort,20,20);          /* Move text pointer in
window */
  Text(Window->RPort,"Hello Sir",9); /* set hello in the window
*/
  /*
  ** Loop until the user hits the close gadget!
  */
  for (;;)
    {
    WaitPort(Window->UserPort);  /* wait for a message */
    message = (struct IntuiMessage *)GetMsg(Window->UserPort);
    if( (message->Class) == MENUPICK ) /* is it a menu
message?*/
      {
      for (i=0; i < 30;i++) str[i] = ' ';
      if( (message->Code) != MENUNULL)
        {
        M0 = MENUNUM(message->Code);
        I0 = ITEMNUM(message->Code);
        S0 = SUBNUM(message->Code);
        sprintf(str,"Menu %x, Item %x, Sitem  %x ",  M0, I0,
S0);
        Move(Window->RPort, 5, 20);
        Text(Window->RPort, str, 24);

        };
```

```
      }
    else break;  /* Not Menu selection, assume close gadget*/
    Process_Item(M0, I0, S0);  /* Process Here */
    ReplyMsg((struct Message *)message);        /* Reply to
message */

    };
  /*
  ** Cleanup when we get the close gadget hit
  ** release everything in the reverse of how
  ** we allocated it.
  */
  ReplyMsg((struct Message *)message);        /* Reply to
message */
  ClearMenuStrip(Window);                     /*Remove Menu*/
  CloseWindow(Window);                        /*close window
*/
  CloseScreen(Screen);                        /*remove the
screen */
  exit(TRUE);                                 /*Good Bye */

  }
/* stub for acutal Menu processing code */
void Process_Item(M0,I0,S0)
int M0;   /* Menu Number     */
int I0;   /* Item Number     */
int S0;   /* Sub Item Number */
  {
  switch (M0)
    {
    case -1 :   /* Close up the shop,this is the close gadget
*/
    printf("Stub Close up Shop\n");
    break;
    default :   /* easy way out */
    printf("Stub for %d:%d:%d\n",M0,I0,S0);
    break;

    };

  }
```


ef'fi-cà-cy, *n.*

power to produce effects or intended results.

Not everyone needs the ability to typeset complex mathematical equations or complicated tables. Many can do without the accents and special characters required for foreign languages. People even exist who don't want a powerful macro facility in their document production system. Ligatures and kerns are overkill for some documents. But your documents are more than simple letters to Mom.

Some folks shun included IFF/ILBM images or ARexx compatibility. A few are willing to hyphenate by hand, or accept poorly spaced lines. And some folks will accept the jaggies—output at less than the full resolution of their printer.

But the quality of your documents is important to you, and your typesetting needs are more than the average fellow's. You need

AmigaTEX

If you are serious about putting words on paper, write for your free demo disk. Move up to the power of AmigaTEX.

√i Radical Eye Software

Box 2081 • Stanford, CA 94309 • BIX: radical.eye

Circle 152 on Reader Service card.


```
FROM LIB:c.o+"Listing_1.o"+"Listing_3.o"
TO "Listing_1"
LIB LIB:lc.lib LIB:amiga.lib
BATCH

MENU,Workbench
ITEM,Open,,
ITEM,Close,,
ITEM,Duplicate,,
ITEM,Rename,,
ITEM,Info,,
ITEM,Discard,,
MENU,Disk
ITEM,Empty Trash,,
ITEM,Initiaze,,
MENU,Special
ITEM,Clean Up,,
ITEM,Last Error,,
ITEM,Redraw,,
ITEM,Snapshot,,
ITEM,Version,,
MENU,My_WorkBench!
ITEM,Not Selected,Selected    ,S
ITEM,TEST,,T
ITEM,SUB Menu 1,,
SUBI,Sub Item 1,,1
SUBI,Sub Item 2,,2
SUBI,Sub Item 3,,3
SUBI,Sub Item 4,,4
ITEM,SUB Menu 2,,
SUBI,Sub Item A,,A
SUBI,Sub Item B,,B
SUBI,Sub Item C,,C
SUBI,Sub Item D,,D
```

LISTING 2.MENU

```
/*************************************/
/* Custom Services Menu Builder  c1989 */
/*************************************/
#include <exec/types.h>
#include <intuition/intuition.h>

struct IntuiText IText[] =
  {
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Open"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Close"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Duplicate"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Rename"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Info"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Discard"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Empty Trash"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Initiaze"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Clean Up"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Last Error"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Redraw"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Snapshot"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Version"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Not Selected"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Selected    "},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "TEST"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "SUB Menu 1"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Sub Item 1"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Sub Item 2"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Sub Item 3"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Sub Item 4"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "SUB Menu 2"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Sub Item A"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Sub Item B"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Sub Item C"},
    { 0, 1, JAM2, CHECKWIDTH, 0, NULL, "Sub Item D"}
  };
#define FLAGS  CHECKIT | ITEMTEXT | MENUTOGGLE | ITEMENABLED
```

```
#define TEXT(x) (APTR)&IText[x]

struct MenuItem M3I2[] =
  {
    {&M3I2[ 1], 52,  3,104, 10,FLAGS,0,TEXT(17),NULL   ,
'1',NULL},
    {&M3I2[ 2], 52, 13,104, 10,FLAGS,0,TEXT(18),NULL   ,
'2',NULL},
    {&M3I2[ 3], 52, 23,104, 10,FLAGS,0,TEXT(19),NULL   ,
'3',NULL},
    {NULL     , 52, 33,104, 10,FLAGS,0,TEXT(20),NULL   ,
'4',NULL}
  };

struct MenuItem M3I3[] =
  {
    {&M3I3[ 1], 52,  3,104, 10,FLAGS,0,TEXT(22),NULL   ,
'A',NULL},
    {&M3I3[ 2], 52, 13,104, 10,FLAGS,0,TEXT(23),NULL   ,
'B',NULL},
    {&M3I3[ 3], 52, 23,104, 10,FLAGS,0,TEXT(24),NULL   ,
'C',NULL},
    {NULL     , 52, 33,104, 10,FLAGS,0,TEXT(25),NULL   ,
'D',NULL}
  };

struct MenuItem M0[] =
  {
    {&M0[ 1],  0,  0, 72, 10,FLAGS,0,TEXT( 0),NULL
,NULL,NULL},
    {&M0[ 2],  0, 12, 72, 10,FLAGS,0,TEXT( 1),NULL
,NULL,NULL},
    {&M0[ 3],  0, 24, 72, 10,FLAGS,0,TEXT( 2),NULL
,NULL,NULL},
    {&M0[ 4],  0, 36, 72, 10,FLAGS,0,TEXT( 3),NULL
,NULL,NULL},
    {&M0[ 5],  0, 48, 72, 10,FLAGS,0,TEXT( 4),NULL
,NULL,NULL},
    {NULL   ,  0, 60, 72, 10,FLAGS,0,TEXT( 5),NULL
,NULL,NULL}
  };

struct MenuItem M1[] =
  {
    {&M1[ 1],  0,  0, 88, 10,FLAGS,0,TEXT( 6),NULL
,NULL,NULL},
    {NULL   ,  0, 12, 88, 10,FLAGS,0,TEXT( 7),NULL
,NULL,NULL}
  };

struct MenuItem M2[] =
  {
    {&M2[ 1],  0,  0, 80, 10,FLAGS,0,TEXT( 8),NULL
,NULL,NULL},
    {&M2[ 2],  0, 12, 80, 10,FLAGS,0,TEXT( 9),NULL
,NULL,NULL},
    {&M2[ 3],  0, 24, 80, 10,FLAGS,0,TEXT(10),NULL
,NULL,NULL},
    {&M2[ 4],  0, 36, 80, 10,FLAGS,0,TEXT(11),NULL
,NULL,NULL},
    {NULL   ,  0, 48, 80, 10,FLAGS,0,TEXT(12),NULL
,NULL,NULL}
  };

struct MenuItem M3[] =
  {
    {&M3[ 1],  0,  0, 96, 10,FLAGS,0,TEXT(13),TEXT(14),
'S',NULL},
    {&M3[ 2],  0, 12, 96, 10,FLAGS,0,TEXT(15),NULL     ,
'T',NULL},
    {&M3[ 3],  0, 24, 96, 10,FLAGS,0,TEXT(16),NULL
,NULL,&M3I2[0]},
    {NULL   ,  0, 36, 96, 10,FLAGS,0,TEXT(21),NULL
,NULL,&M3I3[0]}
  };

struct Menu TheMenu[] =
  {
    {&TheMenu[ 1],  0,  0, 88, 10,MENUENABLED," Workbench
",&M0[0]},
    {&TheMenu[ 2], 96,  0, 48, 10,MENUENABLED," Disk ",&M1[0]},
    {&TheMenu[ 3],152,  0, 72, 10,MENUENABLED," Special
",&M2[0]},
    { NULL       ,232,  0,120, 10,MENUENABLED," My_WorkBench!
",&M3[0]}
  };

struct Menu *MyMenu = &TheMenu[0];
```

LISTING 3.C

```
/*
Menu Builder V2.0
This program takes a simple sequential file and
generates the Menu structures for a program.

Input:

MENU,Name
ITEM,Name,[Select],[Command]
SUBI,Name,[Select],[Command]

Position and Flags will be set to a reasonable default.
Your program  will need to include the statement:

extern struct Menu *MyMenu;
        and
SetMenuStrip(window ptr,MyMenu);

Also, before you close your window you must call:

ClearMenuStrip(window ptr);

or risk a GURU visit!
*/
#include <libraries/dos.h>
#include <stdio.h>
#include <intuition/intuition.h>
#include <proto/dos.h>
#include <stdlib.h>
#include <string.h>
#include <proto/intuition.h>
#include <ctype.h>
#define  CHAR_WIDTH  8      /* used to calculate pixels */
#define NO_ERROR         0  /* used to indicate error   */
#define ERROR_NO_COMMA   1
#define ERROR_BAD_TEXT   2
#define ERROR_NO_MENU    3
#define ERROR_NO_ITEM    4
#define ERROR_NO_DATA    5
#define ERROR_BAD_NUMBER 6
#define ERROR_BAD_COMMAND 7
#define ERROR_TOO_LONG 8
#define MAX_LINE  132      /* input line length max        */
char cline[MAX_LINE];      /* input line                   */
char *name1;               /* File name pointer            */
int cwidth;                /* width of menu item           */
int  lcnt;                 /* total input line count       */
int  scnt;                 /* current line count           */
struct IntuiText *CText;   /* Current Text Item pointer    */
struct IntuiText *AllText; /* All text items               */
struct MenuItem  *CItem;   /* Current Menu Item            */
struct MenuItem  *CSItem;  /* Current Sub Menu Item        */
struct Menu      *CMenu, *AllMenu; /* Menu pointers        */
/*
** prototypes
*/
void Do_Arguments(int, char **);
char * Scan(char *, char, int);
int GetLine(BPTR, char *);
int Process_Menu(char *);
int Process_Item(char *, int);
struct IntuiText *Process_Text(char *);
void Do_Output(void);
int Index(struct IntuiText *);
void Do_Out_Sub(int, int);
/*
** Main routine
*/
void main(argc, argv)
int   argc;                    /*number of arguments
```

```
*/
char  *argv[];                  /*array of ptrs to arg strings
*/
  {
  FILE  *file1;                     /* input File pointer
*/
  int  i;
  int result,error;
  char *cptr;
  /* get the file names and open the files */
  /* initialize input buffers and variables*/
  AllText = NULL;
  CText   = NULL;
  CMenu   = NULL;
  CItem   = NULL;
  CSItem  = NULL;
  AllMenu = NULL;
  error   = 0;
  lcnt = 0;
  /*
  ** Do all files
  */
  for (name1 = NULL, i=1; i<argc && name1 == NULL; i++)
  if ( *argv[i] != '-') name1 = argv[i];
    {
    scnt = 0;
    file1       = fopen(name1,"r");  /*open file for reading*/
    if ( file1 == NULL )
      {
      printf(" Cannot get file open: %s\n",name1);
      exit(10);

      };
    /*
    ** Do All Lines in the file
    */
    while (  ( cptr = fgets(cline,MAX_LINE,file1) ) != NULL)
      {
      scnt++;
      lcnt++;
      cptr = Scan(cptr,' ',FALSE);  /*skip leading blanks*/
      if( *cptr == ';' )continue;   /*skip comments in file*/
      if(       strncmp("MENU",cptr,4) == 0 )
        {
        cptr = Scan(cptr,',',TRUE);
        result = Process_Menu(cptr);

        }
      else if ( strncmp("ITEM",cptr,4) == 0 )
        {
        cptr = Scan(cptr,',',TRUE);
        result = Process_Item(cptr,0);  /* do an Item */

        }
      else if ( strncmp("SUBI",cptr,4) == 0 )
        {
        cptr = Scan(cptr,',',TRUE);
        result = Process_Item(cptr,1);  /* do a Sub Item */

        }
      else  /* error, bad command */
        {
        error++;
        result = ERROR_BAD_COMMAND;

        };
      if( result != NO_ERROR )
        {
        error++;
        fprintf(stderr," Line %d(%s) has syntax error(%d):
        %s\n",scnt,name1,result,cptr);

        };

      };
    fclose(file1);

    };
  /*
  ** all files are read, so output the source if no errors
  */
  if( error == 0 ) Do_Output();
  fprintf(stderr," %d lines input, %d errors\n",lcnt,error);

  }
char *Scan(ptr,c,control)
char *ptr;
char c;
int control;
  {
  while ( *ptr != NULL && *ptr != '\n'
  && ( ( control && *ptr != c ) || ( ! control && *ptr == c) )
)
  ptr++;
  if( *ptr == '\n')*ptr = '\0';
  return( ptr );  /* return the new pointer */

  }
int Process_Menu(ptr)
char *ptr;
  {
  char *tptr;
  int len;
  struct Menu *last;
  if( AllMenu == NULL )
    {
    /* first menu item */
    AllMenu = ( struct Menu *)calloc(1,sizeof(struct Menu));
    CMenu = AllMenu; /* Set current pointer */

    }
  else
    {
    /* add new one to list */
    last = CMenu;
    CMenu = (struct Menu *)calloc(1,sizeof(struct Menu));
    last->NextMenu = CMenu;

    };
```


Lons Fonts Vol. 1

A collection of seven 3D font sets in the Interchange format. Each set has complete upper/lower case letters, punctuation, and numbers! If you're into video or do animations, you need these fonts! $29.95

Momentum Check

A full featured checkbook management package that makes checkbook management easy. Class codes allow you to track any expense you wish. Use standard reports or create your own custom reports. Reconciliation is so easy! $29.95

Momentum Mail

An easy-to-use mailing list management program. Why fiddle with 300-page manuals and spend hundreds of dollars when it can be as easy and affordable as this! $29.95

TeleTutor

An interactive telecommunications tutorial. Everything about telecommunications in one place! Has a simulated BBS to practice uploading and downloading. $29.95

Uzzi Interface

A joystick/mouse interface with an auto-fire rate of 30 rounds/sec!! Switch between auto and transparent mode. 4 ft. extension cable. Blow your game scores away! $34.95

Available at fine dealers, or order direct. Make check or money order payable to:



Micro Momentum, Inc.
P.O. Box 372
Washington Depot, CT 06794
(800) 448 7421

Dealer Inquiries Invited

Please add $1.50 for S&H. C.O.D.s add additional $2.50. All products 90 day warranty.

If you've got a product, we're interested. Give us a call. Amiga is a registered trademark of CBM.

Circle 188 on Reader Service card.

```
  /* Build Menu structure */
  if( *ptr != ',')return(ERROR_NO_COMMA);
  ptr++;
  tptr = Scan(ptr,' ',TRUE); /* find end of string */
  len  = tptr-ptr;
  if( len > 23 )return(ERROR_TOO_LONG);
  CMenu->MenuName = (BYTE *)calloc(1,(unsigned)len+1);
  (void)strncpy(CMenu->MenuName,ptr,len);
  ptr +=len;
  *ptr = '\0';  /* make sure it is a proper string!*/
  CMenu->NextMenu = NULL;
  CMenu->FirstItem= NULL;
  CItem = NULL;
  CMenu->Width = len;
  return(NO_ERROR);

  }
int Process_Item(ptr,type)
char *ptr;
int type;    /* 0 = ITEM, 1 = SUB ITEM */
  {
  struct IntuiText *Process_Text();
  struct MenuItem *ThisItem;
  struct MenuItem *lastI;
  struct MenuItem *lastSI;
  if( CMenu == NULL )              return(ERROR_NO_MENU);
  if( type == 1 && CItem == NULL )return(ERROR_NO_ITEM);
  if( *ptr != ',' )                return(ERROR_NO_COMMA);
  cwidth = -1;
  if( type == 0 )
    {
    /* do an item building a MenuItem */
    lastI = CItem;
    CSItem= NULL;  /* clear current Sub Item Flag */
    CItem = (struct MenuItem *)calloc(1,sizeof(struct Menu-
Item));
    if( lastI == NULL )
      {
      /* First Item for Menu */
      CMenu->FirstItem = CItem;  /* link it to the menu */
      CItem->MutualExclude = -1;
      }
    else
      {
      lastI->NextItem = CItem;   /*link it to the item list*/

      };
    ThisItem = CItem;   /* build it */
    cwidth = CMenu->FirstItem->MutualExclude;
    }
  else
    {
    /* Do a Sub Item */
    lastSI = CSItem;
    CSItem = (struct MenuItem *)calloc(1,sizeof(struct
MenuItem));
    if( lastSI == NULL )
      {
      /* First Sub Item for Menu Item */
      CItem->SubItem = CSItem; /* link in the sub items */
      CSItem->MutualExclude = -1;

      }
    else
      {
      lastSI->NextItem = CSItem;

      };
    ThisItem = CSItem;
    cwidth = CItem->SubItem->MutualExclude;

    };
  /* Process name select and command */
  ptr++;
  ThisItem->ItemFill = ( APTR )Process_Text(ptr);
  if( ThisItem->ItemFill == NULL )return(ERROR_BAD_TEXT);
  ptr = Scan(ptr,',',TRUE); /* find next comma */
  if( *ptr != ',')return(ERROR_NO_COMMA);
  ptr++;
  /* Select name? */
  ThisItem->SelectFill = ( APTR )Process_Text(ptr);
  ptr = Scan(ptr,',',TRUE); /* find next comma */
  if( *ptr != ',')return(ERROR_NO_COMMA);
  ptr++;
  /* Command ? */
  ThisItem->NextItem      = NULL;
  ThisItem->SubItem       = NULL;
  ThisItem->Command       = *ptr;
  ThisItem->MutualExclude = 0L;
  if( type == 0 )
    {
    CMenu->FirstItem->MutualExclude = cwidth;  /* save Item
width */
    }
  else
    {
    CItem->SubItem->MutualExclude = cwidth;    /* save Sub Item
width */
    };
  return(NO_ERROR);

  }
struct IntuiText *Process_Text(ptr)
char *ptr;
  {
  struct IntuiText *dptr, *lasText;
  char *tptr;
  if( *ptr == ',')return(NULL);
  lasText = AllText;
  while( lasText != NULL)
    {
    CText = lasText;
    lasText = lasText->NextText;

    };
  lasText = CText;
  CText = (struct IntuiText *)calloc(1,sizeof(struct Intu-
iText));
  CText->NextText = NULL;
  if( AllText == NULL )
    {
    AllText = CText;

    }
  else
    {
    lasText->NextText = CText;

    };
  tptr = Scan(ptr,',',TRUE);
  CText->IText = (UBYTE *)calloc(1,(unsigned)(tptr-ptr+1));
  (void)strncpy(CText->IText,ptr,tptr-ptr);
  CText->IText[tptr-ptr] = '\0';
  if( cwidth < tptr-ptr )cwidth = tptr-ptr;
  /* check for duplicate, eliminate this one if duplicate */
  dptr = AllText;
  while ( dptr != CText )
    {
    if ( strcmp(dptr->IText,CText->IText) == 0 )
      {
      free( (char *)(CText->IText) );
      free( (char *) CText         );
      CText = dptr;
      lasText->NextText = NULL;

      }
    else
      {
      dptr = dptr->NextText;

      };

    };
  return(CText);

  }
void Do_Output()
  {
  char nxt,quote;
  char Mptr[14];
  int Menu, Item, left, top, width, height;
  int stop;
```

```
  char rest[30];
  /* Output the IntuiText structure first */
  quote = '"';
 printf("/*************************************/\n");
  printf("/* Custom Services Menu Builder  c1989 */\n");
 printf("/*************************************/\n");
  printf("#include <exec/types.h>\n");
  printf("#include <intuition/intuition.h>\n");
  CText = AllText;
  if( AllText != NULL)
    {
    printf("\nstruct IntuiText IText[] =\n");
    printf("  {\n");

    };
  while( CText != NULL ) /* for all text to output */
    {
    if( CText->NextText == NULL)
      {
      nxt = ' ';

      }
    else
      {
      nxt = ',';

      };
    printf("    { 0, 1, JAM2, CHECKWIDTH, 0, NULL,
%c%s%c}%c\n",
    quote,CText->IText,quote,nxt);
    CText = CText->NextText;

    };
  if( AllText != NULL )
    {
    printf("  };\n");

    };
  printf("#define FLAGS  CHECKIT | ITEMTEXT | MENUTOGGLE |
ITEMENABLED\n");
  printf("#define TEXT(x) (APTR)&IText[x]\n\n");
  CMenu = AllMenu;
  Menu = 0;
  left = 0;
  height = 10;
  while ( CMenu != NULL )
    {
    Item = 0;
    CItem = CMenu->FirstItem;
    while ( CItem != NULL)
      {
      CSItem = CItem->SubItem;
      cwidth = CSItem->MutualExclude;
      if ( CSItem != NULL ) Do_Out_Sub(Menu,Item);
      Item++;
      CItem = CItem->NextItem;

      };
    Menu++;
    CMenu = CMenu->NextMenu;

    }
  CMenu = AllMenu;
  Menu = 0;
  while ( CMenu != NULL )
    {
    Item = 0;
    CItem = CMenu->FirstItem;
    cwidth = CItem->MutualExclude;
    CSItem = CItem->SubItem;
    printf("struct MenuItem M%d[] = \n",Menu);
    printf("  {\n");
    top  = 0;
    width =  cwidth * CHAR_WIDTH;
    while ( CItem != NULL )
      {
      Item++;
      CSItem = CItem->SubItem;
      if( CItem->NextItem == NULL )
        {
        sprintf(Mptr,"NULL   ");
        nxt = ' ';

        }
      else
        {
        sprintf(Mptr,"&M%d[%2d]",Menu,Item);
        nxt = ',';

        };
      for(stop=0; stop < 30; stop++)rest[stop] = '\0';
      if( CItem->SelectFill == NULL)
        {
        sprintf(rest,"NULL    ,");

        }
      else
        {
        sprintf(rest,"TEXT(%2d),",Index((struct IntuiText
*)CItem->SelectFill));

        };
      if( !isalpha(CItem->Command) && !isdigit(CItem->Command))
        {
        sprintf(&rest[9],"NULL,");

        }
      else
        {
        sprintf(&rest[9]," '%c',",CItem->Command);

        };
      if( CSItem == NULL )
        {
        sprintf(&rest[14],"NULL");

        }
      else
        {
        sprintf(&rest[14],"&M%dI%d[0]",Menu, (Item-1) );

        };
      printf("    {%s,%3d,%3d,%3d,%3d,FLAGS,0,",
      Mptr,left,top,width,height);
      printf("TEXT(%2d),%s}%c\n",Index((struct IntuiText
*)CItem->ItemFill),rest,nxt);
      CItem = CItem->NextItem;
      top   = top + height + 2;

      };
    printf("  };\n\n");
    CMenu = CMenu->NextMenu;
    Menu++;
```


Are you intimidated by Databases?
Too busy to read a huge manual?

That's why you need Mail-O-Dex.

Mail-O-Dex is a full featured, easy to use mail merge manager. But it doesn't stop there! It also combines the features of a RoloDex, a Phone Dialer, a Label Printer, AND *directly* supports most popular Word Processors. With easy to use VCR style controls, you probably won't even need the easy to understand manual. You can search any or ALL fields, and define field titles. Label sizes and spacing are adjustable.

Best of all, it's only $49.95!!



Available at a dealer near you or order direct. Send check or money order payable to

Dealer inquiries welcome

KarmaSoft
P.O. Box 1034
Golden, Co. 80401
(303) 277-1241

Circle 124 on Reader Service card.

Attention Readers!



AC WANTS YOU!

Since we at Amazing Computing™ cannot determine the dependability of our advertisers from their ads alone, we want your feedback. If you have had a problem with an advertiser in AC™, let us know! Send a complete description of your exchange(s) with the advertiser—along with the names of the individuals involved—and we'll do our best to get to the bottom of things.

Ad Complaints
PiM Publications, Inc.
Amazing Computing
P.O. Box 869
Fall River, MA 02722


```
  };
CMenu = AllMenu;
printf("struct Menu TheMenu[] =\n");
printf("  {\n");
left = 0;
top  = 0;
Menu = 0;
while ( CMenu != NULL )
  {
  Menu++;
  width = (CMenu->Width + 2)  * CHAR_WIDTH;
  if( CMenu->NextMenu == NULL )
    {
    sprintf(Mptr,"NULL       ");
    nxt = ' ';

    }
  else
    {
    sprintf(Mptr,"&TheMenu[%2d]",Menu);
    nxt = ',';

    };
  (void)strncpy(&rest[1],CMenu->MenuName,CMenu->Width);
  rest[0] = ' ';
  rest[CMenu->Width+1] = ' ';
  rest[CMenu->Width+2] = '\0';
  quote = '"';
  height = 10;
  printf("
{%12s,%3d,%3d,%3d,%3d",Mptr,left,top,width,height);
  printf(",MENUENABLED,%c%s%c,&M%d[0]}%c\n",quote,rest,quote
  (Menu-1),nxt);
  left = left + width + CHAR_WIDTH;
  CMenu = CMenu->NextMenu;

  };
printf("  };\n\n");
printf("struct Menu *MyMenu = &TheMenu[0];\n");
}
int Index(ptr)   /* Find the position index */
struct IntuiText *ptr;
  {
  int idx;
  idx = 0;
  CText = AllText;
  while  ( CText != NULL && ptr != CText)
    {
    idx++;
    CText = CText->NextText;

    };
  return(idx);

  }
void Do_Out_Sub(Menu,Item)
int Menu, Item;
  {
  int SItem;
  int stop, width, sleft;
  int top, height;
  char Mptr[14];
  char rest[30];
  char nxt;
  top = 0;
  height = 10;
  printf("struct MenuItem M%dI%d[] = \n",Menu,Item);
  printf("  {\n");
  width = CMenu->Width*CHAR_WIDTH;
  SItem = 0;
  while ( CSItem != NULL )
    {
    for(stop=0;stop<30;stop++)rest[stop] = '\0';
    SItem++;
    if( CSItem->NextItem == NULL )
      {
      sprintf(Mptr,"NULL     ");
      nxt = ' ';

      }
    else
      {
      sprintf(Mptr,"&M%dI%d[%2d]",Menu,Item,SItem);
      nxt = ',';

      };
    if( CSItem->SelectFill == NULL )
      {
      sprintf(rest,"NULL   ,");

      }
    else
      {
      sprintf(rest,"TEXT(%2d),",Index((struct IntuiText
*)CSItem->SelectFill));

      };
    if( !isalpha(CSItem->Command) && !isdigit(CSItem->Command))
      {
      sprintf(&rest[8],"NULL,");

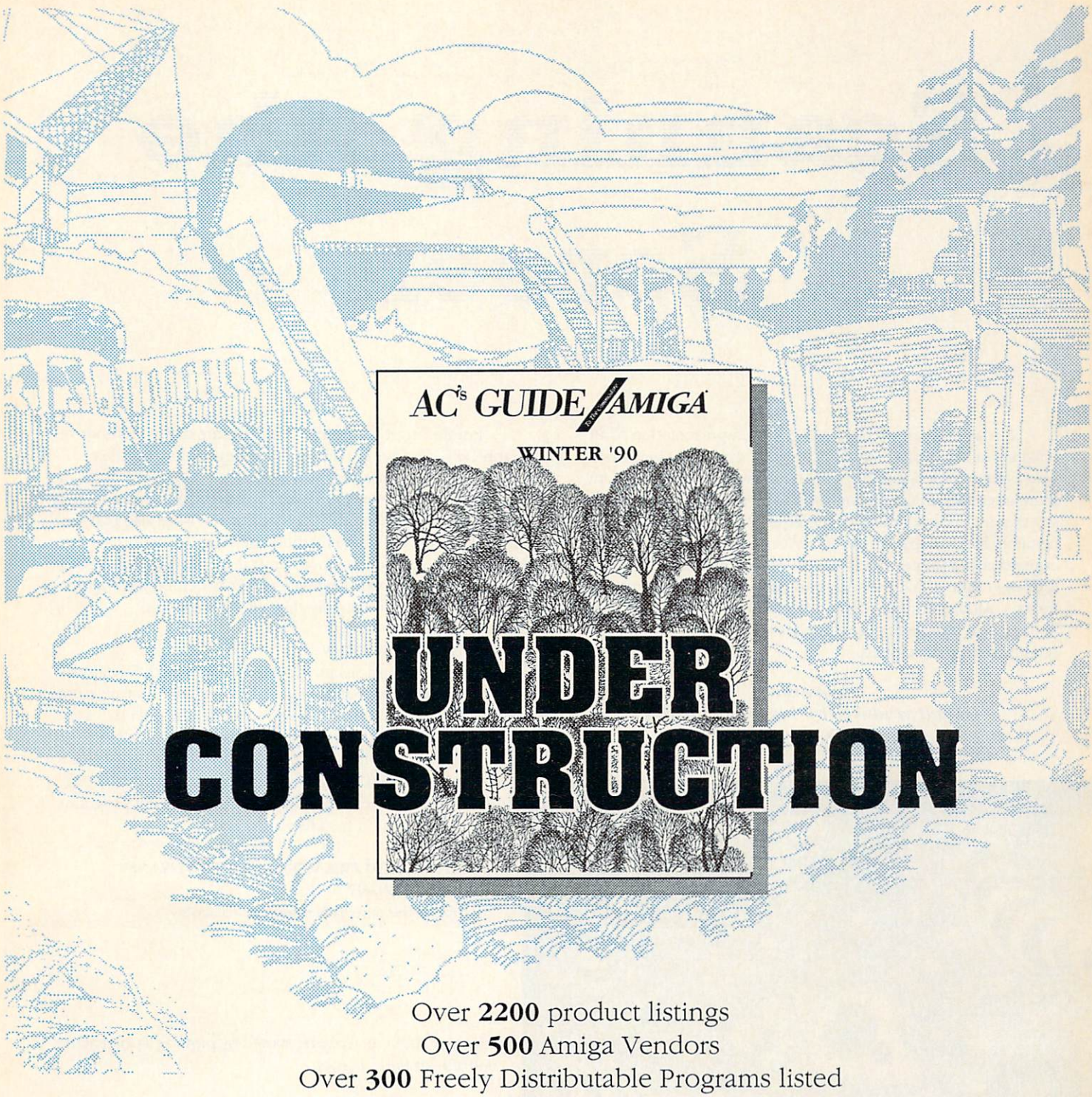
      }
    else
      {
      sprintf(&rest[8]," '%c',",CSItem->Command);

      };
    sprintf(&rest[13],"NULL");
    sleft =  (width * 5)/10;
    stop  =  top + 3;
    printf("    {%s,%3d,%3d,%3d,%3d,FLAGS,0,"
    ,Mptr,sleft,stop,width,height);
    printf("TEXT(%2d),%s}%c\n",Index((struct IntuiText
*)CSItem->ItemFill),rest,nxt);
    CSItem = CSItem->NextItem;
    top = top + height;

    };
  printf("  };\n\n");

  }
```

•AC•



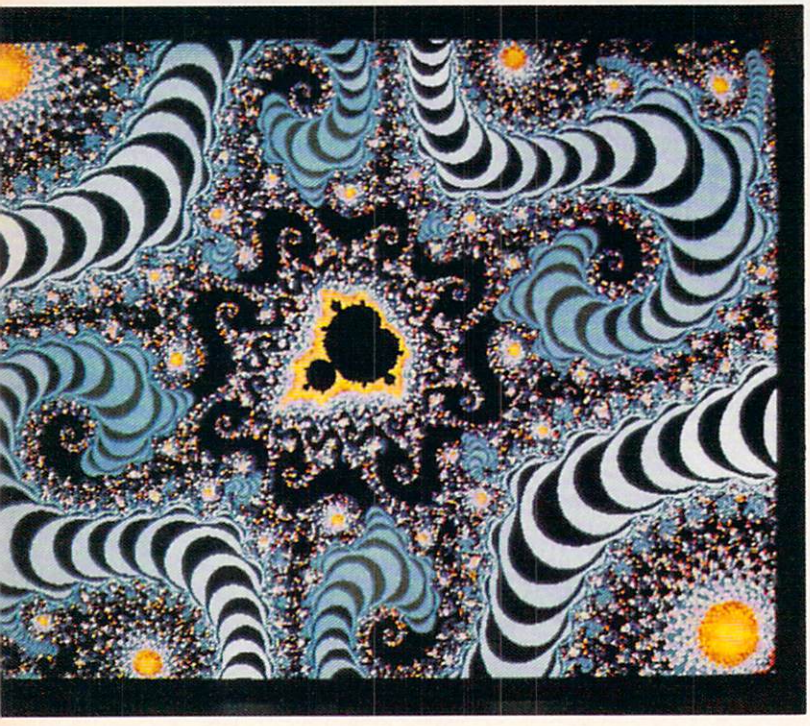
Over **2200** product listings
Over **500** Amiga Vendors
Over **300** Freely Distributable Programs listed

AC's Guide To The Commodore Amiga Winter '90
Coming soon to your local Amazing Dealer or order today!
The Guide to a successful Amiga Winter

Scanning the Screen

Mathematicians explain a graphical representation of an equation as a map of its solutions, and that it is used to help them gain a better understanding of the mathematics. However, I am more interested in the picture itself as an artform. That's not to say that an understanding of the equations is not valuable. It certainly helps, especially in the never ending search for new equations and for new methods of displaying them. However, it doesn't take a mathematician to have a lot of fun drawing fractals. All that is needed are a few practical skills and some patience. In this article one such skill, called scanning the screen, will be developed.

Figure One:
Output produced by this month's listing (640 x 400 Pixel Mode)



Before I start let me say that there are many methods of graphically displaying mathematical equations and these articles are not intended to be a systematic treatment of all of them. Instead, I develop the one method that is particularly valuable for displaying the Mandelbrot set. Note also that this article deals only with the method, the series of steps that must be programed in order to draw a fractal. It is still too early to talk about the underlying mathematical theory behind the Mandelbrot set, although eventually this will be discussed.

SCANNING THE SCREEN

In previous articles the development of each fractal was observed pixel by pixel, from left to right, and row by row, from bottom to top. That's what is called scanning the screen. It's the systematic plotting (illuminating) of all of the pixels on the drawing surface. It should come as no surprise that there is a standard way of doing this, and once it has been learned, it can be applied over and over again to different programs that draw different fractals.

ONE HUNDRED TWENTY-EIGHT THOUSAND POINTS OF LIGHT

Suppose the screen is scaled as follows:

```
xmin = -10
xmax = 10
ymin = -10
ymin = 10
```

Scanning the screen can be started by plotting a point at the bottom left hand corner.

```
PSET (FNx(xmin),FNy(ymin))
```

The variables xmin and ymin each store the value -10, the Cartesian coordinates of the bottom left hand corner of the drawing surface, as specified by the above scaling. Those values get passed to the coordinate conversion functions FNx() and FNy(), which calculate their corresponding pixel numbers (Amiga coordinates) and then pass them to the PSET command. Refer to Article II in this series if more explanation is needed.

Continuing with the systematic scanning of the screen, the very next pixel needs to be plotted along the bottom

Part Four in the Fractals Series

by Paul Castonguay

row. To accomplish this, the pixel's horizontal coordinate must be calculated (the vertical coordinate remains the same because it is in the same row).

THE NEXT PIXEL

In every example so far, usually on the third line, the variable dx, which represents the width of one pixel, is calculated using xmin, xmax, and the number of pixels across the screen. Remember that all the pixels are arranged side by side, thus forming a long row. So, to get to the next pixel, simply move to the right by an amount equal to the width of one pixel, dx. Plot the point:

```
PSET (FNx(xmin+dx),FNy(ymin))
```

Well, that was simple enough, but what about the next pixel after that? How about:

```
PSET (FNx(xmin+dx+dx),FNy(ymin))
```

OK then, what about the next 629 pixels along the entire row? FNx(xmin+dx+dx+dx+dx+dx+dx...etc) will not do, it requires too much typing. AmigaBASIC's "FOR NEXT" loop construction should be used.

```
FOR i=xmin to xmax STEP dx
    PSET(FNx(i),FNy(ymin))
NEXT i
```

In the above loop PSET gets executed 632 times, once for every pixel along the bottom row. The first time through, the variable 'i' has the value xmin, -10. Each time through the loop 'i' gets increased by the value of dx. That's what the 'STEP dx' means. In this example the value of dx is 0.0316957211, but it is not necessary to know that. dx is always referred to symbolically, by its name. The Amiga will handle the actual value. That's what computers are for! The value of dx will be different whenever you scale the screen to different Cartesian coordinates. Continuing, each time through the loop the value of 'i' is passed first to the coordinate conversion functions, and then to the PSET command, thus plotting every pixel along the row. The loop executes over and over again, continually increasing the value of 'i' in steps of 'dx', until finally 'i' exceeds xmax. At that point the loop ends and the computer goes on to execute whatever program lines follow the 'FOR-NEXT' construction in the program.

Here is a complete program which does the above task and nothing more.

```
DEF FNx(x) = INT(((x-xmin)+dx/2)/dx)
DEF FNy(y) = 195 - INT(((y-ymin)+dy/2)/dy)

xmin = -10
xmax = 10
ymin = -10
ymax = 10

dx = (xmax-xmin)/631
dy = (ymax-ymin)/195

SCREEN 1, 640, 200, 4, 2
WINDOW 2, , (0,0)-(631,186), 0, 1

FOR i=xmin TO xmax STEP dx
    PSET(FNx(i),FNy(ymin))
NEXT i

WINDOW CLOSE 2
SCREEN CLOSE 1
```

Note that in previous examples in the earlier articles an extra dx/2 was added, as follows:

```
FOR I=xmin to xmax+dx/2 STEP dx
```

The reason for this is to guarantee that the last pixel along the row doesn't get accidentally left out. All computers store numbers with a limited accuracy. For example, the number 2/3 gets stored as 0.6666667 which is a little larger than it should be (the last digit is rounded up to a seven). The term dx is calculated from (xmax-xmin)/631, and depending on the values of xmax and xmin its accuracy could also be questionable. It could also be a little larger than it should be. As the loop executes, dx gets added to 'i' 631 times until it's supposed to equal xmax. Well, suppose on the 631'th time the value of 'i' actually overshoots xmax by a little bit, say by 0.000001. That would be

enough to cause the loop to stop without plotting the very last pixel in the row. The extra dx/2 term guarantees that this doesn't happen.

The program has been carefully written to be consistent with examples in this series of articles. That way previously learned skills can be built on.

MORE ROWS

So much for one row of pixels. Now, what about the whole screen? Well, that's not too much harder. Replace the loop in the above program with the following:

```
FOR j=ymin TO ymax+dy/2 STEP dy
    FOR i=xmin TO xmax+dx/2 STEP dx
        PSET(FNx(i),FNy(j))
    NEXT i
NEXT j
```

This is called a nested loop, one loop inside the other. The inside loop draws the normal horizontal line. But now there is an outside loop in which 'j' represents the vertical position of that line. The value of 'j' starts at ymin and increases by the amount of 'dy' until it finally exceeds ymax+dy/2. For each value of 'j' a complete horizontal line gets drawn at a new vertical position. Thus every pixel on the screen gets illuminated. Run the previous program using this nested loop in place of the single loop ran before and confirm that the entire screen gets plotted, pixel by pixel.

DRAWING FRACTALS

Since starting these articles two skills have been developed which can be applied to drawing fractals, scaling the screen (from Article II), and now, scanning the screen. But more is needed to produce a fractal, the magic of an equation is required. It's the equation that tells the program what color to plot each pixel, thus producing something more pleasing than a blank white screen. Looking at the screen scanning loops in the previous articles, two very important lines inserted before the PSET command can be seen.

```
GOSUB Calculate
GOSUB Select.Color
PSET(FNx(i),FNy(j))
```

Every time through the scanning loop, for every pixel on the entire screen, a calculation is performed, presumably based somehow on the Cartesian coordinates of that pixel. Then, depending on the results of that calculation, a color is selected. Finally, the pixel is plotted in the chosen color.

HOW ABOUT AN EQUATION?

Since this article is about the method, and not the mathematics, let's choose an equation that will be easy to understand.

$$x^2 + y^2 = 25$$

Easy to understand means, understanding the calculations that are specified; it is not necessary to know the graph the calculation will produce. If the actual values for x and y were given, x = 3 and y = 4, it should be easy to perform the calculations and determine if the values satisfy the equation.

$$3^2 + 4^2 = 25$$

So x and y values satify the equation and are a solution.

But if x = 2 and y = 3 were given, then

$$2^2 + 3^2 \quad 25$$

So the x and y values do not satisfy the equation and are not a solution.

INEQUALITIES

Often knowing if the result of a calculation is greater than or less than the value specified in the equation can be important. The algebraic notation for such an equation could be something like this:

$$x^2 + y^2 < 25$$

Such equations are known as inequalities. If the values x = 2 and y = 3 are given, the result of the left side of the equation will equal 13, and since 13 is less than 25, these values cause the equation to be true.

MASTER PLAN

Suppose the value of x in the above inequality represents the horizontal coordinate of any pixel on the screen, and y the vertical (Cartesian coordinates). The equation could be calculated using those coordinates, tested to determine if that pixel caused the inequality to be true or false, and finally, the pixel could be colored accordingly (one color for true and another for false). If this were done for every pixel on the screen, a picture representing the trueness or falseness of the inequality for every pixel on the entire screen would be obtained. It would sort of be a picture of the equation. Here's what we will do:

1- Decide the range of Cartesian coordinates.
2- Scale the screen.
3- Scan the screen
4- Calculate the equation using the cartesian coordinates eachPixel
5- Test the result for true or false.
6- Color pixels according to result.

Below is a program which uses the screen scanning method to graphically represent the above equation.

```
DEF FNx(x) = INT(((x-xmin)+dx/2)/dx)
DEF FNy(y) = 195 - INT(((y-ymin)+dy/2)/
dy)

xmin = -10
xmax =  10
ymin = -10
ymax =  10


dx = (xmax-xmin)/631
dy = (ymax-ymin)/195

SCREEN 1, 640, 200, 4, 2
WINDOW 2, , (0,0)-(631,186), 0, 1

FOR j=ymin TO ymax+dy/2 STEP dy
    FOR i=xmin TO xmax+dx/2 STEP dx
        GOSUB Calculate
        GOSUB Select.Color
        PSET(FNx(i),FNy(j))
    NEXT i
NEXT j

WINDOW CLOSE 2
SCREEN CLOSE 1

END

Calculate:
    Result = i * i  +  j * j
RETURN

Select.Color:
    IF Result < 25 THEN
        COLOR 8
    ELSE
        COLOR 10
    END IF
RETURN
```

Take a look at the 'Calculate' subroutine. The 'x' and 'y' of the original mathematical equation have been replaced by 'i' and 'j'. The i's and j's come from the screen scanning part of the program, and they represent the horizontal and vertical positions of pixels on the screen (in Cartesian coordinates). Those are the numbers that will be used to evaluate the equation every time the 'Calculate' subroutine is executed. The calculation is stored in the variable 'Result'. The RETURN statement sends the computer back up to the screen scanning loop to continue with the next statement.

The next statement in the screen scanning loop is the 'Select.Color' subroutine. Here the value of 'Result' is compared

THINKER
Hypertext
for AMIGA
"..stunning capabilities..simple to operate.." "..superbly crafted.." - Gary Gehman, Amiga Sentry, 6/89
Version 2.0 with ARexx



Hypertext extended to Picture files, ARexx ports, and applications. Version 2 adds more keyboard controls, database functions, display fonts, and ARexx. Thinker acts as a database for ARexx programs, sends messages to ARexx ports, and more. Write, design, think, plan. HYPERTEXT!

Upgrade $20 No Credit Cards
$80 Add $5 for COD; CA res. add tax; 30 day guarantee

Poor Person Software
3721 Starr King Circle, Dept 5
Palo Alto, CA 94306
(415)-493-7234

Circle 127 on Reader Service card.

width have been scaled to have equal dimensions, xmin = -10, xmax = 10, ymin = -10, and ymax = 10. As a result, the pattern is wider than it is tall, it looks squashed. Change the scaling to:

```
xmin = -10
xmax =  10
ymin = -7
ymax =  7
```

Now run the program again. With the ratio corrections the program produces a circle.

Now, change the single line in 'Calculate' to:

```
Result = i*i*i*i + j*j*j*j
```

and the first line of 'Select.Color' to:

```
IF Result < 625 THEN
```

and run it again. It is a different pattern. This is a graph of the equation:

$$x^4 + y^4 < 625$$

The above method is a complete plan or algorithm for producing graphic images of mathematical equations. It is the same method that has been used in the previous articles to draw fractals. The above example uses a simple equation so as to not distract from learning the method. Naturally the mathematics behind the Mandelbrot set is a little more involved and the lines in the 'Calculate' and 'Select.Color' subroutines are correspondingly more complex. But the rest is the same.

SOFTWARE PLUS

Chicagoland's *Amiga* software, hardware, and repair stores

Staffed by Amiga users

Full selection of books, magazines & peripherals, including imported software

Wheeling 708-520-1717
Chicago 312-338-6100
Hanover Pk 708-837-6900

Circle 143 on Reader Service card.

Templicity™
100 Ready-made Templates for Your Amiga Spreadsheet!
Available for Analyze!, MaxiPlan, Superplan, and VIP Professional
Personal Financial Planning
Home Office and Small Business
New! 1989 Federal Income Taxes
Real Estate and Loan Analysis
Home and Business Budgets
Accounting and Bookkeeping
Save Time and Money
Many other unique worksheets included!
Full User Support
30 day, no risk, money-back guarantee!
Complete pkg. of 100 templates $34.95
To order, send check or money order to:
The Sterling Connection
Box 4850
Berkeley, CA 94704
Specify which spreadsheet you have.
To order by phone, call (415) 655-2355
Mastercard and Visa accepted.
CA res. add 6% ($2.10) sales tax.
(Free shipping)

Circle 157 on Reader Service card.

against the number 25. The IF-THEN-ELSE construction is used to separate the pixels into two groups, those that represent solutions to the equation (true) and those that don't (false). When 'Result' is less than 25 (true), the computer executes the line following the word 'THEN'. When 'Result' is not less than 25 (false), the computer executes the line following the word 'ELSE'. The command 'END IF' marks the end of the construction. Thus each pixel is assigned a color register depending on the result of the previously executed 'Calculate' subroutine.

Run the above program and see what happens. The equation should graph as a circle, but why does the graph appear in an oval?

ASPECT RATIO

The computer screen is not a perfect square; it is a rectangle. The aspect ratio is the technical term which describes the relative difference between its height and width. Get a ruler and confirm that the height is about seven inches and the width is ten. In the above program the height and

INTERLACE MODE (640 x 400 PIXELS)

The second article of this series mentioned that there were actually four coordinate systems available on the Amiga. Thus far the 640 x 200 pixel mode has been used, because it is the best compromise between graphic detail, memory requirement, and the time required to produce a fractal. However, for the fun of giving the Amiga a tough job, this month's example is done in both 640 x 200 and 640 x 400 pixel resolutions. It is possible to do either by modifying only one line of the program. For 640 x 200 pixel mode:

```
Vertical.Pixels = 200
```

For 640 x 400 pixel mode (also called interlace mode):

(continued on page 100)

The Amazing Computing Freely Redistributable Software Library announces the addition of...

New Orleans Commodore Klub's

inNOCKulation Disk

Version 1.5b

To help inform Amiga users of the newer Amiga viruses and provide them with the means to detect and eradicate those pesky little critters!

Files and directories on the inNOCKulation Disk include:

Virus_Texts (dir)
Various text files from various places (Amicus #24, PeopleLink, and elsewhere!) describing the Virus(es) and people's experiences and their recommendations; TVSB "The Virus Strikes Back": satirical text describing future efforts to rid the universe of the dreaded (silicon) viruses! Interview with the alleged SCA virus author!

WB_VirusCheckers (dir)

VirusX4.0
Runs in the background and checks disks for viruses or non-standard boot blocks whenever they are inserted. (Recognizes several viruses and non-standard boot blocks. Removes virus in memory.) Finds the new Xeno file virus.

Sentry
Revision of VirusX1.01 in Lattice C.

ViewBoot
Highly active mouse-driven disk and memory virus-checker which allows you to look at the pertinent areas (useful in case you suspect a NEW virus!)

VRTest3.2
Watches memory for viruses; will alert the user and allow their removal if found. Can check & INSTALL disks, etc.

CLI_VirusCheckers (dir)

XenoZap
Disables the Xeno virus in infected files. Checks every file in all directories of the specified devices allowing the fast removal of the Xeno virus.

AVirusII
From The Software Brewery (W. German). Disables a virus in memory.

Clk_Doctor3
Corrects problems with the clock (caused by malignant programs, perhaps not really a "virus") (A500 & A2000)

Guardian1.1
Checks for attempts at viral infection at boot! Allows you to continue with a normal boot (if desired). Includes a small utility program to permanently place the program on a copy of your Kickstart disk.

KillVirus
Removes (any?) virus from memory.

VirusKiller
A graphically appealing and user friendly program by TRISTAR.

Boot-Block_Stuff

SafeBoot2.2
SafeBoot will allow the user to save custom boot sectors of all your commercial disks and save them for such an emergency. If a virus somehow manages to trash the boot sectors of a commercial disk, just run SafeBoot and it will restore the boot sectors, therefore saving your disk!!

Virus_Alert V2.0.1
Yet another anti-virus program with a twist. Once installed on your boot disk a message is displayed just after a warm or cold boot notifying the the user that the disk and memory are virus-free, and forcing a mouse-button press before continuing.

BootBack1
Saves and restores boot-blocks. Runs from CLI only.

Antivirus *aka AVBB*
Includes SEKA assembler source.

XBoot
Converts a boot-block into an executable file, so you may use your favorite debugger (Wack, Dis, ...) to study it.

The inNOCKulation disk also includes icons and arc files.

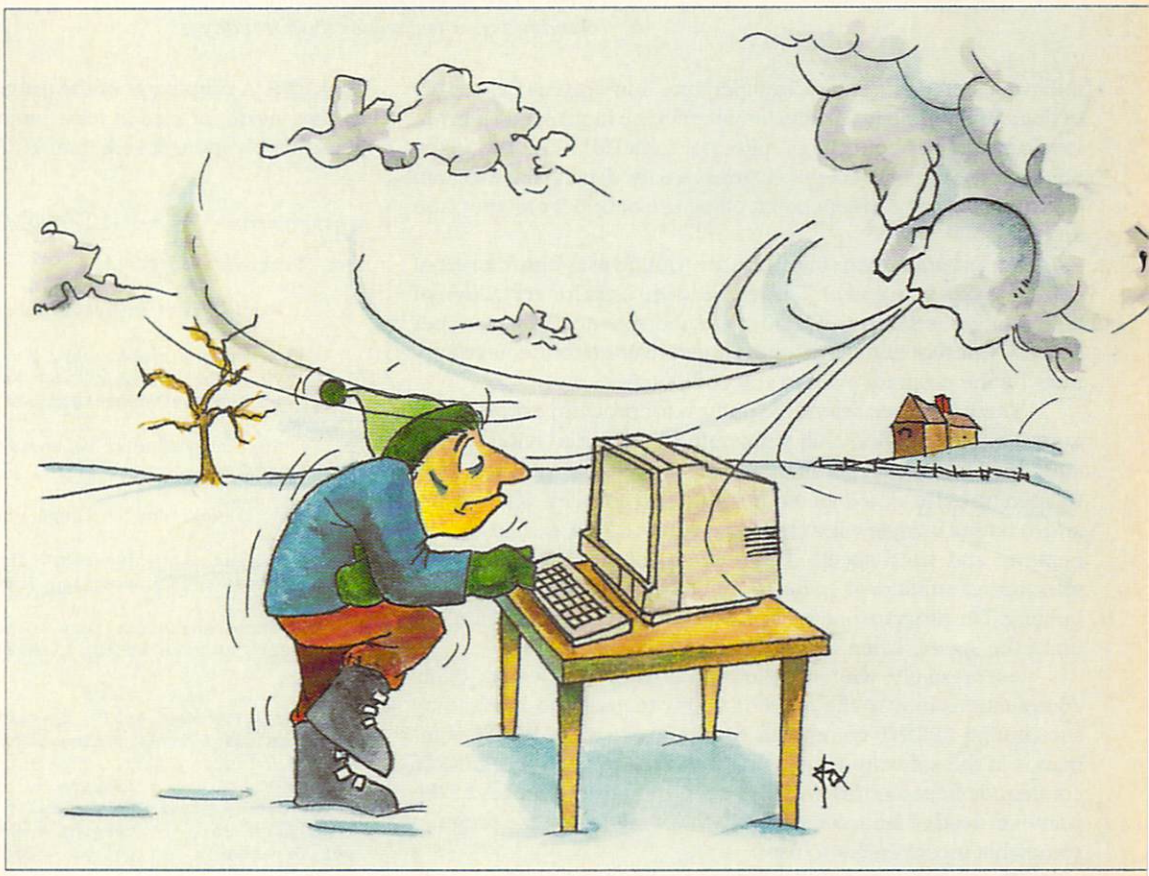
To order the inNOCKulation disk, send: **$6.00** *includes postage & handling*

($7.00 for non-subscribers)

Amazing Computing
inNOCKulation disk orders
P.O. Box 869
Fall River, MA 02722

IT'S COLDER THAN YOU THINK!

by Robert Klimaszewski



As you listen to the weather reports on the radio or evening news this winter, you may hear the announcer say something like, "the temperature outside is five degrees; but the wind chill temperature is a frigid twenty-four below." Have you ever wondered where that second number comes from? (Especially when you actually step outside and realize how intensely cold it is when the icy wind hits you in the face!) This article will describe a simple program which calculates the wind chill temperature based on a choice of inputs.

Hill's Comfort Factor is a measure of the amount of energy removed from a person's skin because of convection currents caused by winds. Since the number of "calories of energy lost per square centimeter of skin surface per second" is not a very convenient (or useful) measure for most people, something much more familiar will be used. Weather announcers quote an effective temperature which is based on the present air temperature and the wind speed. This quantity is called the "wind chill temperature."

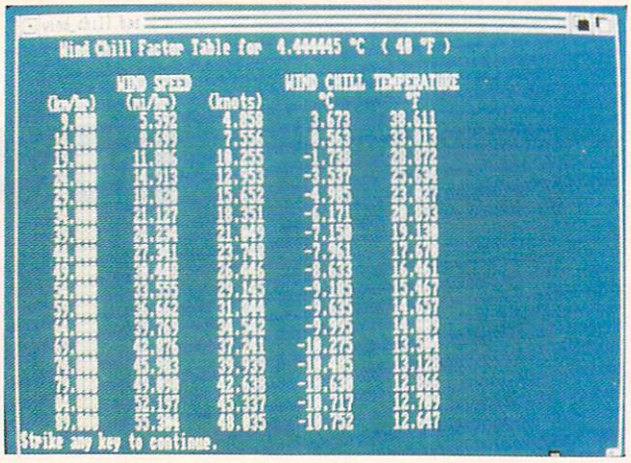
The wind chill temperature (or wind chill factor, as it is frequently — and incorrectly — called) is determined from a nonlinear correction to the air temperature. The correction depends on wind speed and on the temperature of still air. Some more sophisticated equations also include the effects of relative humidity, but since weather announcers rarely provide the humidity during the colder months of the year, the program given here is based only on the first two factors.

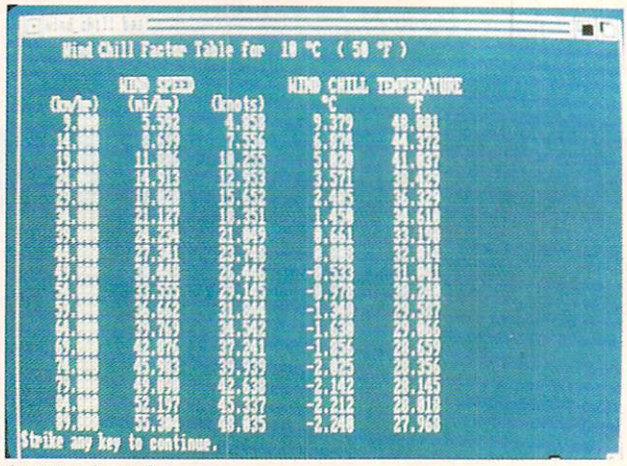
The AmigaBASIC program, "Wind_Chill.BAS," calculates the wind chill temperature, W, using the simple formula:

$$W = T0 - (T0 - T)(a + bv^{1/2} + cv)$$

where T is the temperature of still air (as measured by a thermometer) and v is the wind speed. The reference temperature T0 is approximately that of human skin. (Skin temperature is actually a little warmer than the value used for T0 because the latter has been modified to partially account for humidity of the air and skin moisture.) The remaining three numbers (a, b, and c) are semi-empirical constants, and depend only on the units chosen for the variables v, T, and W.

For convenience, the AmigaBASIC program was written to calculate the wind chill temperature based on the user's choice of units for the input values. Therefore, it is necessary for the user to





Brrr... Two screen shots showing how the wind makes it feel colder than it really is.

indicate whether the entered temperature is in degrees Celsius (C) or degrees Fahrenheit (F), and/or whether the input speed is given in kilometers per hour (K) or miles per hour (M). (To obviate the need for the user to remember unnecessary details, the program was designed to be independent of the use of upper or lower case letters.)

For programming simplicity, the routine uses only one set of constants (T0, a, b, and c). The temperature used for evaluation of the equation is given in Celsius, and the speed is in meters per second. Consequently, the inputs are converted to the necessary units by the program prior to any computations.

There are three selectable options for program execution: 1) output a table of wind chill temperatures for varied wind speeds and constant air temperature; 2) output a table of wind chill temperatures for varied air temperature and constant wind speed; and 3) output a single wind chill temperature for a specified air temperature and wind speed. The third option also allows for the selection of additional pairs of input values, if desired, before quitting. The program output is provided in both metric and English units; the speed, when outputted, is also given in knots.

As presently written, Wind-Chill sends the results of its computations only to the monitor. This can easily be changed by substituting LPRINT commands for the appropriate PRINT commands in the software code. (Be careful not to change all PRINT commands because this will also send the input requests to the printer instead of the screen! Clearly, this would make the program somewhat inconvenient to use.)

There are two small limitations to the program Wind-Chill. The value for the temperature must be in the range -31°F to +59°F; and the value for the wind speed is restricted to the range of 5.6 mph to 55.9 mph. If a mistake is made, and a value is entered outside these ranges, Wind-Chill will politely inform the user of the error, reminding the user of the limitations, and it will ask for a new value to be inputted.

It is important to keep in mind that the output of this program may not coincide exactly with the predictions of a weather forecaster. A forecaster has access to more detailed information from the National Weather Service, and may use more complex equations to determine the wind chill temperature. In any case, Wind-Chill will provide a reasonable estimate of the effective outdoor temperature when the radio D.J. casually mentions that the wind is gusting up to 50 mph, and the home thermometer reads only 10°F. A simple use of the program will tell you that it's better to stay inside, or else at least prepare yourself for temperatures considerably more severe than 10°F!

LISTING ONE

```
'
'    Program Wind_Chill
'
'      Wind Chill Temperature Computation
'
'    Program copyright 1988, R A Klimaszewski
'      released to the public domain for non-commercial use
'      no fee may be charged for the distribution of this
program
'      This notice must be included with any distributed copy
of the program
'
 DEF FNchill(vel,temp) = temp0 - (temp0 - temp)*(a + b*SQR(vel)
+ c*vel)
 DEF FNcelsius(fahren) = five.ninths#*(fahren - 32#)
 DEF FNfahren(celsius) = nine.fifths#*celsius + 32#
'
'  For above equation: temp in *C, vel in m/s
'    -35*C <= T <= +15*C; 2.5m/s <= vel <= 25m/s [9km/hr,90km/
hr]
'
 temp0 = 33! : a = .4517 : b = .4322 : c = -.04322
 five.ninths# = 5#/9# : nine.fifths# = 1.8#
 v1 = 3.6 : v2 = 2.237 : v3 = 1.943
 true = -1          : false = 0
 enter.data = true : done = false
 temp.lo = -35!     : temp.hi = +15!   ' *C
 vel.lo = 2.5       : vel.hi = 25!     ' m/sec
'
 PRINT "  Determination of Wind Chill Temperatures"
 PRINT
 PRINT "    Input/Output temperatures are *F and *C"
 PRINT "    Input/Output velocities are km/hr and mi/hr"
 PRINT
 determine.wind.chill = true
Cycle.start:
 WHILE (determine.wind.chill)
   PRINT
   PRINT " Possible Wind Chill Temperature Calculations:"
   PRINT
   PRINT "    A.  Table of Wind Chill Temperatures vs. Velocity
for Fixed Temperature"
   PRINT "    B.  Table of Wind Chill Temperatures vs.
Temperature for Fixed Velocity"
   PRINT "    C.  Wind Chill Temperature for Specific Input
Temperature and Velocity"
   PRINT "    Q.  Quit"
```

```
   PRINT
   INPUT "SELECT INPUT FORMAT (A/B/C/Q): ", format$
   PRINT
   IF (UCASE$(format$) = "A") THEN
     calculate = enter.data
     WHILE (calculate)
       INPUT "ENTER TEMPERATURE and TEMP SCALE (T,[C]elsius/
[F]ahrenheit): ", temp, t.type$
       IF (UCASE$(t.type$) = "F") THEN
         temp.f = temp
         temp.c = FNcelsius(temp)
       ELSE
         temp.c = temp
         temp.f = FNfahren(temp)
       END IF
       IF ((temp.c < temp.lo) OR (temp.c > temp.hi)) THEN
         PRINT "The valid temperature range is -35*C <= T <=
+15*C"
         PRINT "  or -31*F <= T <= 59*F.  Try again!"
         PRINT
         calculate = enter.data
       ELSE
         PRINT
         PRINT
         PRINT
         PRINT "      Wind Chill Factor Table for "; temp.c;
"*C  ("; temp.f; "*F )"
         PRINT
         PRINT "              WIND SPEED             WIND CHILL
TEMPERATURE"
         PRINT "    (km/hr)     (mi/hr)     (knots)       *C
*F"
         FOR vel.m = vel.lo TO vel.hi STEP 1.388888889#
           chill.c = FNchill(vel.m,temp.c)
           PRINT USING "#######.###"; v1*vel.m, v2*vel.m,
v3*vel.m, chill.c, FNfahren(chill.c)
         NEXT vel.m
         calculate = done
       END IF
     WEND
   END IF
   IF (UCASE$(format$) = "B") THEN
     calculate = enter.data
     WHILE (calculate)
       INPUT "ENTER WIND SPEED and SPEED SCALE (V,[K]m per hr/
[M]i per hr): ", speed, s.type$
       IF (UCASE$(s.type$) = "K") THEN
         vel.km = speed
         vel.m = speed/v1
         vel.mi = vel.m*v2
       ELSE
         vel.mi = speed
         vel.m = speed/v2
         vel.km = vel.m*v1
       END IF
       IF ((vel.m < vel.lo) OR (vel.m > vel.hi)) THEN
         PRINT "The valid wind speed range is 9km/hr <= V <=
90km/hr"
         PRINT "  or 5.6mi/hr <= V <= 55.9mi/hr.  Try again!"
         PRINT
         calculate = enter.data
       ELSE
         PRINT
         PRINT
         PRINT
         PRINT "  Wind Chill Factor Table for "; vel.km; "km/hr
( "; vel.mi; "mi/hr )"
         PRINT
         PRINT "      ACTUAL TEMPERATURE    WIND CHILL
TEMPERATURE"
         PRINT "        *C          *F          *C          *F"
         FOR temp = temp.lo TO temp.hi STEP 5!
           chill.c = FNchill(vel.m,temp)
           PRINT USING "#########.##"; temp, FNfahren(temp),
chill.c, FNfahren(chill.c)
         NEXT temp
         calculate = done
       END IF
     WEND
   END IF
   IF (UCASE$(format$) = "C") THEN
     calculate = enter.data
     WHILE (calculate)
       need.speed = true : need.temp = true
       WHILE (need.temp)
         INPUT "ENTER TEMPERATURE and TEMP SCALE (T,[C]elsius/
[F]ahrenheit): ", temp, t.type$
         IF (UCASE$(t.type$) = "F") THEN
           temp.f = temp
           temp.c = FNcelsius(temp)
         ELSE
           temp.c = temp
           temp.f = FNfahren(temp)
         END IF
         IF ((temp.c < temp.lo) OR (temp.c > temp.hi)) THEN
           PRINT "The valid temperature range is -35*C <= T <=
+15*C"
           PRINT "  or -31*F <= T <= 59*F.  Try again!"
           PRINT
           need.temp = true
         ELSE
           need.temp = false
         END IF
       WEND
       WHILE (need.speed)
         INPUT "ENTER WIND SPEED and SPEED SCALE (V,[K]m per
hr/[M]i per hr): ", speed, s.type$
         IF (UCASE$(s.type$) = "K") THEN
           vel.km = speed
           vel.m = speed/v1
           vel.mi = vel.m*v2
         ELSE
           vel.mi = speed
           vel.m = speed/v2
           vel.km = vel.m*v1
         END IF
         IF ((vel.m < vel.lo) OR (vel.m > vel.hi)) THEN
           PRINT "The valid wind speed range is 9km/hr <= V <=
90km/hr"
           PRINT "  or 5.6mi/hr <= V <= 55.9mi/hr.  Try again!"
           PRINT
           need.speed = true
         ELSE
           need.speed = false
         END IF
       WEND
       chill.c = FNchill(vel.m,temp.c)
       PRINT
       PRINT
       PRINT
       PRINT "      Wind Chill Factor for "; temp.c; "*C  ( ";
temp.f; "*F )"
       PRINT "        and "; vel.km;"km/hr  ("; vel.mi; "mi/hr
)"
       PRINT
       PRINT "   WIND CHILL TEMPERATURE =";
       PRINT USING "####.##"; chill.c;
       PRINT " *C  (";
       PRINT USING "####.##"; FNfahren(chill.c);
       PRINT "*F )"
       INPUT "Do you wish to enter a new temperature and wind
speed"; test$
       IF (UCASE$(test$) = "Y") THEN calculate = enter.data
ELSE calculate = done
     WEND
   END IF
   IF (UCASE$(format$) = "Q") THEN
     determine.wind.chill = false
   ELSE
     PRINT "Strike any key to continue."
     WHILE (INKEY$ = "")
     WEND
   END IF
 WEND
Exit.pt:
 END
```

•AC•

Thinker

by Robert Klimaszewski

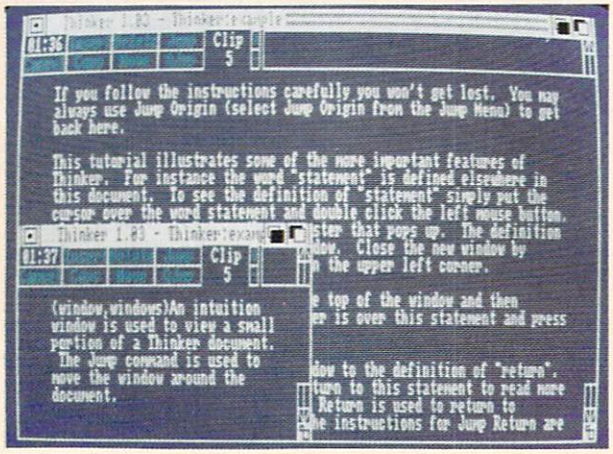


Figure 1: Thinker lets you open multiple windows on the same or different documents.

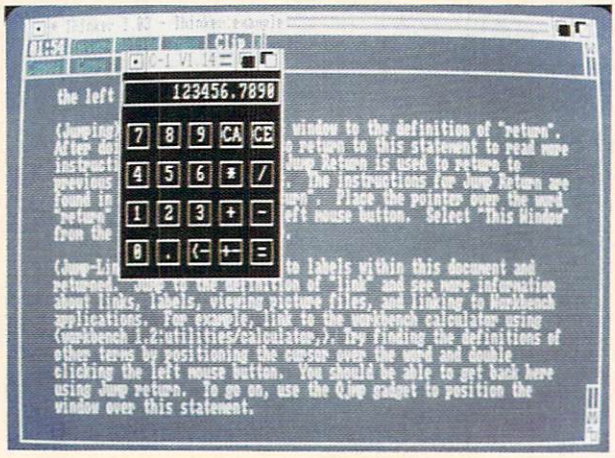


Figure 2: The Workbench Calculator window was opened using the bracketed link in the Thinker document.

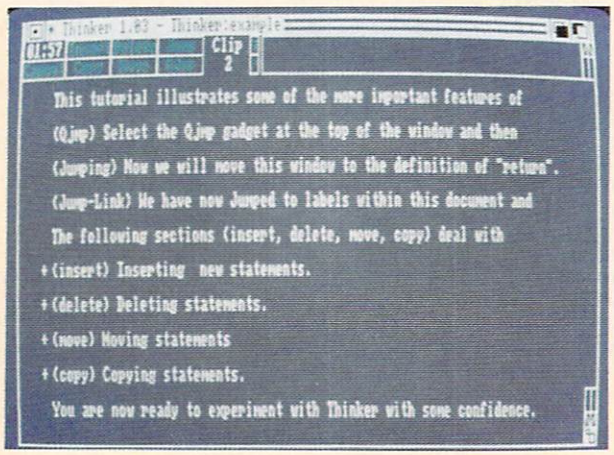


Figure 3: By setting the "clipping level" to 2, several sublevels of the document are concealed from direct view, enabling you to scan quickly through a project.

Although Thinker can be described in simplest terms as an outline processor (OP), it is far more sophisticated than that. In fact, the program, which is the first Amiga program based on "hypertext," is much more flexible than any other OP, including some word processors (WP). The author of Thinker refers to the program as an Idea Processor, and that is a very apt description.

The manual, tutorial "example," and other samples on the program disk display only a hint of the potential of Thinker. The program is remarkably useful; it can be used to create outlines, textual descriptions, instruction manuals, dictionaries, phone directories, project schedules, computer programs, etc. In addition, it is ideally suited for creation of on-line presentations since it is possible to incorporate both sound and graphics into the body of the text via "links" to other files. Consequently, Thinker can be used to create multi-media presentations and "documents."

Two additional features of Thinker are capabilities that are normally associated with a WP, rather than an OP, but serve to enhance the value of this unique program. To supplement its basic outlining capabilities, Thinker includes a spelling lexicon and an indexer. And another highly unusual feature is its ability to execute external programs.

HYPERTEXT

Hypertext has received great publicity lately due to a highly successful implementation on the Apple Macintosh, called HyperCard. Consequently, prior to reviewing the actual program, a brief discussion of the hypertext system and its early history seems to be in order.

The basic concepts of hypertext are derived from the ideas of Vannevar Bush, President Franklin Roosevelt's science advisor, for an on-line data reference system using microfiche and photocells. His dreams were never realized, but they served as the foundation for the research of Douglas Engelbart and Ted Nelson, some twenty years later.

Engelbart developed the system called Augment which is an on-line work environment used for memos, notes, communication, and project design. Augment was designed to be a hierarchal system which permits nonhierarchal branching. And to make the system fast and easy to use, Engelbart invented a clever new computer input device called the mouse. (Thanks, Doug.)

Nelson, on the other hand, created Xanadu which is an ever-expanding system for publishing, text storage and text retrieval. The term "hypertext" was coined by Nelson to describe the nonhierarchal text generated by Xanadu. But his program was intended to serve as an electronic link to many other forms of media as well (e.g., artwork and movies). In essence, Xanadu was designed to be a highly sophisticated database management system (DBMS), but one that differs significantly from the DBMS's that are popular today.

As a DBMS, hypertext connects various pieces of data by the use of "links." Consequently, the structure of hypertext as a DBMS is unique. The organization of data is basically hierarchal, like an outline, rather than repetitious as in a conventional DBMS.

In a normal outline there are primary topics, and there are subtopics that are subordinate to those primary topics, The connection between the levels is established by the use of alphabetic and numeric labels for each topic. Skimming through only the primary topics gives a brief overview of the document subject. For more detail a person would read to lower levels; the most specific information would be provided at the lowest level of the outline. There are occasions, however, when a single sublevel might reasonably belong to several higher levels; in such situations, a nonhierarchal connection method, such as a tree structure, might be more efficient than an "outline structure."

With hypertext, the connections between levels and sublevels can be quite nonlinear. (Levels and sublevels are sometimes refered to as "nodes" in discussions of hypertext.) The advantages of such a system can be considerable, as one piece of information can be accessed by many different subjects. By use of the hypertext links, a user can easily circumvent the obvious external ordering of a document to satisfy his own needs. (This nonlinear access to a sublevel is something that is required often as many different kinds of texts or manuals are read. How many times have you read a document that said something like, "See section X below?") The capability of hypertext to use links yields a very efficient data storage structure in which the links become the fundamental building blocks. The situation can be considered as analogous to a program built up from several orders of subroutines. (A forth program is probably the closest program structure).

In hypertext, different sections of a document are seperated using "labels". The form of a label depends on the particular hypertext implementation, but usually it is a textual string, rather than an alphabetic or numeric character as in a typical outline. The level can then be accessed from anywhere in the hypertext document where the label appears embedded in the text, not merely from the immediately preceeding higher level. With Hypertext, the connection to the stated reference is immediate by linking directly to the label. After the appropriate material has been read, a simple "return" brings the reader back to the original topic, either directly or by back-tracking.

PROGRAM USAGE

Thinker creates a "project" which consists of numerous "statements" arranged in a hierarchy, with links to move around. As pointed out earlier, those links may be to points within the same project or to some other external file or executable program.

A "statement", as defined in Thinker, is the equivalent of a level in an outline. However, a statement, which starts with a label, may consist of numerous sentences, thus being more like a paragraph in a document. Therefore, a considerable amount of information can be presented at each level of the document. Many statements will also include sublevels that define the hierarchal ordering of ideas and provide a greater degree of detail. A statement and all statements that are subordinate to it form a complete branch.

The statement label (located at the beginning of a statement and enclosed in parentheses, "()") is a textual string, consisting of alphanumeric and special characters. Hopefully, the string label will be logically selected to provide a general indication of the contents of the particular level. These labels are the anchors to which Thinker links as movement occurs through the Thinker project, "jumping" from one level to another.

Any word (or group of words enclosed in angle brackets, "<>") in a statement is a potential link to another statement. If that word (or group of words) is also the label to a statement, then the word is a real link. In fact, the link can be text into another file, sound, graphics, to executable programs (including "Workbench" utilities, such as Notepad) in the same or a different drawer, or even on a separate disk.

Thinker also allows the use of several labels, separated by commas on one line, to mark a statement. This increases system flexibility and provides a project author with some added freedom when creating the links to various sublevels. Defining the name of the link for a target audience is made much easier. Consequently, the users of a particular project can move about the document more easily because variants on a phrase may yield the same link (e.g., graphic, graphics, picture).

The ability to move between different projects via the link system is also very useful and powerful. It is a convenient means of joining different types of projects together.

The most obvious type of interfile link is to another text file. Thinker even provides a clever browsing mechanism called "See-Thru" which allows one to directly view the contents of another project as part of the current database. Changes can be made to affect one or both files simultaneously.

As pointed out in the introduction, projects created for on-line use with Thinker can make use of various media. This feature is also handled via interfile links; and it is certainly the most powerful aspect of the system. Clearly, the use of Thinker as a "hypermedia" database system is where this program has its greatest potential. In this regard, Thinker is far superior to a conventional database manager program, even though its lack of a built-in programming language (such as SQL or those used by other relational database managers) makes

Clearly, the use of Thinker as a "hypermedia" database system is where this program has its greatest potential. In this regard, Thinker is far superior to a conventional database manager program...

it less powerful for some applications than a program like Superbase Professional. But then it wasn't really designed to compete with programs like that!

Graphics are linked as easily as text. The bracketed link provides the picture location (disk and drawer) and file name. Clicking on the link then displays the picture in its own window. A click or any keystroke returns the calling window. This is a feature that could be very valuable for the creation of an interactive, on-line manual or document. Both IFF and HAM modes are supported.

Sound files can be executed with the ease of pulling up a picture. The files can be stored using any of several Amiga formats. This feature could be useful for designing a story-board presentation, especially in conjunction with graphics.

menu items are also accessible via text gadgets, which are displayed in the upper left corner of the screen. However, not all of the Command submenu items can be accessed via the gadgets; only the most important ones are provided. A convenient, special gadget appears directly under the cursor when the mouse menu button is double-clicked, no matter where the mouse is positioned on the screen. This provides the user with a good extra that reduces the motion of the mouse all over the desktop, and reminds me of a similar feature on very expensive engineering workstations.

The program also makes extensive use of Requesters for input of data. The Requesters are used for verification of several different actions, such as when a branch is to be moved or deleted. The only annoying requester is the Requester that

Table 1
Evaluation of Thinker

Criteria	Score *	Weight
Documentation	4.5	80
Installation	5.5	60
Copy Protection	5.0	50
Ease of Learning	4.5	80
Use of Amiga Features	5.5	80
Ease of Use	5.5	90
Performance	5.5	100
Versatility	5.5	90
Error Handling	5.0	80
Company Support	5.5	80
Overall	5.2	

*Notes: min: 0 = Extremely poor, max: 6 = Exceptional

• *USING THE PROGRAM* •

USE OF THE INTUITION INTERFACE

Thinker was designed as a mouse-oriented program. As such, it utilizes all features of the mouse interface effectively. All program functions are accessable via mouse selection from the menu; and the menu titles and items are generally obvious in purpose. In addition, the Command pops up whenever jumping to a new link in the file, but it can be turned off optionally.

One of the nicest features of Thinker is the ability to open multiple windows on a project being created or examined. Since it is very common to jump nonsequentially through a document when using hypertext, it is quite convenient to open a new window, keeping the primary text visible while examining the side reference. If it is necessary to view another document briefly, this too, can be done using another window without affecting the main project. Up to eight windows can be opened at a time, so (in principal) eight different Thinker projects can be modified simultaneously (GOOD LUCK!).

The program was also designed for multitasking, but because I have a plain vanilla 512K A1000, I was only able to open tools that did not require much memory. To the limited extent that I utilized the feature there were no problems in this area.

In addition, Thinker provides a mechanism to internally link to an executable program, instead of through the multitasking features of Intuition. The same method that allows linkage to an external picture file also gives a user the ability to execute an external application program, such as the Workbench calculator or even a spreadsheet program. Of course, in order to accomplish this feat, there must be an adequate amount of memory on the Amiga that is running Thinker. Thinker simply won't allow another application to open, if there is not enough memory.

The Amiga Clipboard is also fully supported. The Clipboard is used for all system editing. This can be particularly handy when moving information from one Thinker project to another. Simply open both files in separate windows and use the Edit functions described below to move a sentence, a statement, or even a complete branch from one file to the other via the Clipboard.

OPENING A PROJECT

A document is opened in one of several ways: (1) by using the Create New menu item to start a brand new project, (2) by selecting the Open Project menu item, which brings up a Deluxe Paint-style file requester, (3) by linking to an existing project, or (4) by importing an ASCII document prepared by some other application, such as a text editor or spreadsheet. (Linking to an existing project is an unusual way to open a file, but it is consistent with the normal program operation). New text is entered simply by typing at the line cursor.

MOVING THROUGH A PROJECT

Traversing a Thinker project is not quite as simple as moving through a word processing document. That is, of course, the consequence of creating a hierarchal text with nonhierarchal jumping. To simplify matters, the author of Thinker has provided several mechanisms to move

through a file. The first technique is the most obvious and old-fashioned: use the cursor keys. The page will scroll forward and backward one line at a time as the edge is passed, so this method is the most precise, but it is also the slowest method for moving through large portions of the file at a time.

The technique of scrolling is familiar to anyone who has used even the least expensive Amiga WP. However, Thinker is a little bit different, because the "up" arrow gadget on the right edge of the display is designed for more rapid motion through a project: one statement back at a time. The "down" arrow gadget moves the text more conventionally: one line forward at a time. And there is no vertical scroll bar for rapid motion through the file.

The most flexible means of moving through a project is by use of the Jump command. Moving hierarchally from statement to statement or between levels is easy. Predefined moves from the Jump submenu are used (e.g., Jump Preceding), or a large-scale nonhierarchal jump using the "Jump Link" command. (In a conventional outline program, it may be quite a chore to locate the appropriate sublevel, even if a search routine is available to simplify the process.) There is only one useful option that I feel is missing: Jump End. (...the opposite of Jump Origin). So as a simple trick, simply create the label "<theEnd>", and place it at the end of the file.

Unfortunately, the ability to jump so freely around a project can be a source of problems if the user is not careful. It is easy to get far afield from the original topic by hopping around to follow a thread of thought; without caution, it is possible to lose track of the initial origin. This problem is not unique to Thinker; it is the most serious plague of all hypertext systems. Fortunately, Thinker provides a jump table to assist the user in returning either stepwise or via shortcuts to his point of origin. I also found that judicious use of new windows for the temporary Jump made returning to the origin very simple — just close the new window when done.

When used for the creation of outlines, Thinker has two specialized features that are refered to as "filters" in general hypertext jargon. These filters control what is visible on the screen at any given time. The two filters are for the clipping level and for the line count.

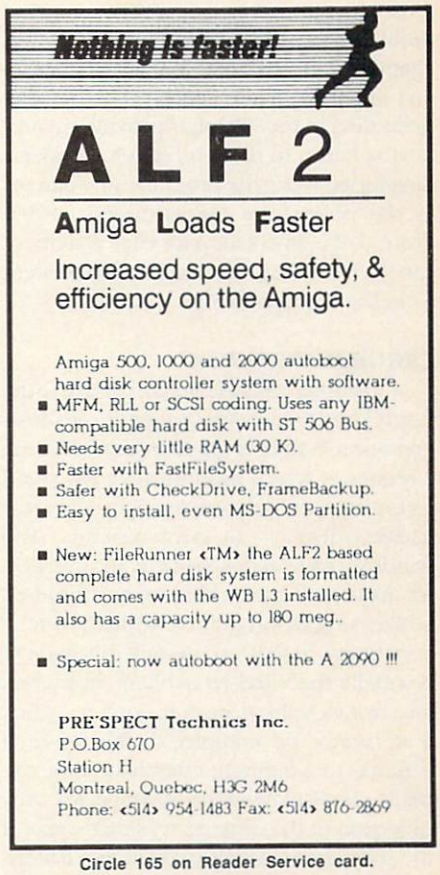
The clipping level is used to control the number of sublevels that are visible at any given time. A clipping level of one, for example, would show only the highest levels (the major headings) in the document; all sublevels would be hidden from immediate viewing. For a clipping level of two, the next level down is also visible, and so on. The clipping level can be set either via the "Clip" item of the Options menu or by use of the "+" and "-" gadgets at the top of the window.

When a level contains hidden sublevels because of the clip setting, a "+" is displayed before the label. Clicking on the "+" opens up the next sublevel down for that particular topic. The plus gadget is then replaced with a minus gadget in front of the label. Selecting the "-" results in the opposite action; the lower sublevel(s) are filtered from view.

Another convenient feature for outline processing is the ability to set the line count. This filter allows the user to selectively display 1, 2, 5 or all of the lines (not sentences) within a statement or level. This is a great advantage when skimming quickly through a document to determine its basic contents. However, reading half a sentence (or less) can sometimes be a little bit cryptic! I found that selecting two statement lines is usually the best way to get a quick overview of a document. Of course, different writing style might be more terse than mine, and one line could be perfectly adequate!

EDITING A PROJECT

There are two ways to edit a Thinker project. The first is conventional: normal editing functions (cut, paste, etc.) are available for modifying or correcting the text within a statement as required. Text that is to be cut or copied is placed into the Clipboard and can then be pasted into a different part of the statement, or it can be pasted into any other project statement. In addition, Insert, Delete, Copy, and Move editing functions are provided to manipulate entire levels and sublevels. Statements, groups (several statements at the same level), and branches can be created, shifted around, duplicated, and deleted with considerable ease. When trying to organize thoughts, this feature is invaluable. For example, while writing this review, I decided upon a major revision of the ordering for several subtopics. It was decidely simple to move the complete branches to their new locations — this included a change in level. The program handled the changes easily. (The only thing that I lost while writing this review was my sanity, because my baby daughter repeatedly tried to remove a microfloppy from the diskdrive with a pencil, whenever I paused to answer the phone.)



Circle 165 on Reader Service card.

No corrections are actually applied to the project until the item "Apply Modifications" in the "Project" menu is selected. Fortunately, this is one function that can be selected from the keyboard, using the obvious key combination: Amiga-U [Update]. There is also a nice reminder that data exists in the computer memory. An asterisk in the program title bar serves as an annunciator to indicate that data should be saved. If the project is closed by selecting the Close gadget in the upper left corner, everything is lost. On the other hand, if Close Project menu item is selected, the program politely warns that modifications have been made but not saved. And then a choice is given to save the modifications.

The spelling checker is a useful aid to editing. A small dictionary (33,000 words) is provided with Thinker. It can be enlarged to include a specialized vocabulary. Or if preferable, the supplied dictionary can be replaced entirely.

PROJECT UPDATES

One of the great problems in updating a program or a document is maintaining records of the changes or the dates when

those changes were made. This very critical problem, known as "configuration management," can be quite difficult when several versions of the same program have been distributed. Thinker provides a very neat solution to this problem for projects developed using the program. All changes to the project are time-stamped, and if desired, the time-stamp for each statement can be viewed by selecting the proper item from the Options menu.

ERROR MONITORING

Thinker has a clever alternative to the usual Amiga requester concerning an error condition in the program. Instead of using a requester which must be acknowledged by the user, a window to the right of the text gadgets displays an error message. The window provided adequate information for nearly all the problems encountered while using the program (and there weren't very many, at that) to prepare this review. Normally the window is blank, but when an error condition occurs, such as a link that cannot be completed, the window changes to a different combination of colors while displaying the message. My only objection to the scheme was that some of the messages were displayed only briefly, and then vanished before they could be acknowledged by a keystroke or mouse click.

EXTRA FEATURES

There are some extra features of Thinker that exemplify the author's concern for the user interface. One of the screen gadgets, for example, is a system clock. Having the time always visible is a nice convenience. In addition, one menu, called Memory, shows the available RAM left at any instant. Since Thinker can utilize a considerable amount of memory for a large project, it is a great advantage to know how much RAM is still available for updates or edits to a file. And last, the Update Icon menu item allows selection of the Options saved as part of the project icon. Consequently, when the project is loaded again, whether for editing or viewing, features such as the clipping level or read cache setting are automatically reinstated in place of the program defaults. And since that data is saved as part of the icon, it can also be altered in the Info screen, if desired, at some other time.

The power of Thinker, as well as its versatility, make it a very valuable and useful program...And never once did the Guru intrude upon my serenity with his anti-kharmic, crimson meditation messages.

The indexing feature is also a very useful capability. A list of all the labels used in the document can be generated quite simply. This is handy for creating an index to browse through the document or for creating a glossary.

THE MANUAL

The Thinker manual is a brief document. It is small (5-1/2" x 8-1/2", 58 pages, comb-bound), but adequately describes the operation of the system and the program functions. Every menu item is described individually. However, it would have been helpful to have either a table of the Amiga key shortcuts (for the functions that have one), or an indication of the keyboard shortcuts with the individual functions.

And the Table of Contents does not provide sufficient information to locate a specific function. An index should have been included in the manual to provide such data. Fortunately, because the manual is so brief, it is not very difficult to locate the information within a few minutes of searching, although it is annoying.

GENERAL COMMENTS

While I discovered several problems in the program which will be discussed below, only one of them is related to a deficiency in the performance of the program. The other comments concern minor "defects" that affect the ease of use of the program, and reflect a personal bias which may not be shared by other users. None of those other problems had any significant bearing on my final rating of Thinker.

DISPLAY CONTROL

As I began to experiment with Thinker, I discovered that typing beyond the end of a screen didn't cause a scroll forward, even though I was still typing the same "statement". Anything that had been typed beyond the end of the screen was banished to the Twilight Zone. It was necessary to implement a "Jump" down to complete the statement.

Another disadvantage was the inability to move the cursor to the right or left by a full word at a time. Many WPs use the shift key in conjunction with the right and left arrows to accomplish this. It is obviously not a required feature, but it would certainly be a desirable convenience.

In the same vein, the "TAB" key doesn't work. If for any reason a tab or indention was needed, then the spacebar must be used to accomplish the task. In an outline processor this is not a serious problem, but I happen to use the TAB key often, even when creating an outline.

I also found the lack of control over the typeface to be annoying. Even in an outline there are reasons to use Boldface, Italic and Underlined typefaces.

[Note: the program does not support the use of multiple Amiga fonts. However, I don't consider that to be a disadvantage. If you need to get that fancy, then you should be using something designed for creating a flashy output; clearly not what Thinker was created for. You can, of course, use this program to write the "rough" draft, and then "Export" the document to an ASCII text file, which in turn could be modified using some WP or print formatter.]

FUNCTION CONTROL

A new user of the program might be surprised by how few of the menu items are selectable from the keyboard. While there are a few places that I would prefer to have complete keyboard control, using either function keys or an <Amiga> key combination, most of the functions require follow up with a mouse action of some kind, so little would be gained by use of a keyboard shortcut.

I have one small complaint about the use of the Delete gadget. The "Confirm" requester would be much better if the branch or group to be deleted was highlighted by reverse type.

There is one final problem that was encountered in the function control area. When I first tried the spell checker, it worked much like expected for such a small lexicon; the consequences of using the spell checker, however, were not at all what I expected from a program that had otherwise proven itself to be a truly superior product. It became the Case of the Vanishing Menus!

After using the checker, the Free menu item was selected to release the extra memory that was used by the function. Subsequently, I discovered a couple of problems, it became impossible to open any new windows even though that option was still provided in the "Jump" requester. And second, several of the menu and submenu lists vanished. The latter problem seemed especially weird since it was still

possible to select the invisible menu items! When I contacted the program author to consult with him on the problem, he was quite surprised but decided that the problem was caused by the limited memory of my 512K system. Consequently, the only real problem that I found in the program reflects a serious limitation in use of the spell checker. Use of the spell checker on an Amiga with only 512K memory will result in peculiar behavior by the system.

DATA ENTRY

I have a habit of trying out any new program on my Amiga before I read the instructions. I consider that to be a partial test of the successful implementation of a good user interface. With Thinker, this turned out to be a serious mistake on my part.

After I followed the directions in the tutorial example for awhile, I decided to experiment on my own. The first thing I tried was a simple entry of a new level into the existing file. It didn't take long to discover that the program performed unbuffered writes to the disk after each LINE or CHARACTER! I had finished a ten-line "Statement" before the program had finished typing the first line! It was at that point that I decided to put away the program, until I had a chance to read the entire manual.

After reading the manual, I learned that the reason the problem occurs is because Thinker is a disk-oriented program. This implies that all data is kept on disk until needed. Since Thinker was designed to work on huge data files, keeping the data on disk is necessary to conserve computer memory. Only a small portion of the project is brought into the computer's memory at a time.

In the "Options" menu there is an item called "Read Cache On." This must be selected to avoid the serious typing lag. The computer's memory is used to buffer all keystrokes instead of writing each one directly to disk; disk accesses are thus required only infrequently. (The manual recommends the use of a disk accelerator program such as FACC-II, if possible, but the Read Cache option is available to those who don't own such a program).

There is an unusual way to undo errors in Thinker, but it is one which must be exercised with care. The "Discard Modifications" menu item is the only method for undoing a serious blunder. Unfortunately, it discards ALL modifications made since the last "Apply Modifications." So get in the habit of saving the project changes frequently, or there won't be a means of correcting minor mistakes. (It can be a hard habit to develop. I've been typing for the last ninety minutes without saving any changes, so I guess this a good time for me to take my own advice. Fortunately, I never make any big mistakes... Oops!! Was that a power glitch?)

PRINTING A THINKER DOCUMENT

The discussion of the Print menu item in the manual points out that it is possible to print out the entire project or only a branch. However, there is little information concerning print formatting methods. Thinker uses the printer port directly, rather than the Preference printer drivers because the program author felt that most people would use a project created with Thinker as the input to a conventional WP or as an interactive, on-line document. Although all keyboard keys can be directly entered into the text (including the <ESC> and <CR> key codes, as well as all the <ALT>-character sequences used for special and foreign characters), the Amiga PRT: port filters out many, if not all, of the escape sequences sent to it. This may negate attempts by the user to obtain printer control. My printer, for example, won't receive most of the printer control commands sent to it directly through the PRT: port. I can't even control page breaks!

FILE BACKUP

For a program as sophisticated as this one is, it seems strange that there is no provision for automatic generation of a backup file. The documentation advises the user of Thinker to make a copy of his project prior to modifying it with Thinker. Other application programs handle this feature automatically, and the author indicates that this feature may be included in a future version in the Options menu, so the user could decide if a backup should be automatically generated or not. Giving the choice to the user might be the best approach, since some Thinker applications might be extremely large, and routine backup generation might result in a filled disk.

CONCLUSION

Thinker is a solid program. It's flexibility far exceeds any other program that I've used on any computer. It has been designed to accommodate many of the possible needs of a user. After using the program for many hours, I have few complaints about the implementation of any function. The power of Thinker, as well as its versatility, make it a very valuable and useful program. This is one program that I will use often. For $80.00, it is a real steal! And remarkably, the Guru NEVER intruded upon my serenity with his anti-kharmic, crimson meditation messages.

A numerical summary of my evaluation of this program is given in Table 1.

•AC•

Create your own fractal worlds with

Scene Generator

- Fractal landscape program
- Fast, fun, easy to use
- Full Intuition interface
- Preview a scene in seconds
- Photographically real results
- Optionally generate clouds
- Complete color control
- Great backgrounds for painting or animation

Natural Graphics
P.O. Box 1963
Rocklin, Ca. 95677
(916) 624-1436

$39.95 ppd US, CA add tax
Demonstration disk $6

Circle 160 on Reader Service card.

Thinker, version 1.03
Poor Person Software
3721 Starr King Circle
Palo Alto, CA 94306
415-493-7234

Price: $80.00
(Demo Disk: $5.00)

Requirements
System: any Amiga (A500, A1000, A2000, A2500)
Memory: 512K (more preferred)
Op Sys: Workbench 1.2 or higher

Copy Protection: None

Did You See This Issue?

December 1989

Features:

"Dear TIME" An open letter to TIME Magazine by Don Hicks
ARexx Part II by Steve Gillmor
"Now that I've ARexx, what do I do with it?"
The MIDI Must Go Thru by Br. Seraphim Winslow
Expanding the Amiga into a full-blown MIDI studio.
Amiga Circuits Hardware Project by John Iovine
Input informnation via the parallel port.
View From The Inside: by Melissa Jordan Grey
A tour of Bars & Pipes' by its musical designer.
The Great Amiga Holiday Games Tip by B. "Santa" Solomon
Santa gives us two hot tips for the Holiday Season.
Commercial Rollout by Jean & Ben Means
"The Computer for the Creative Mind" advertising blitz.
Behind The Scenes by Geoffrey Williams
A look behind Commodores's new $15-million campaign.
AmiEXPO Santa Clara by Rosie O'Grady

Reviews:

Support Your Local Library reviewed by Marion Deland
WordPerfect's desktop utility, WordPerfect Library.
Delphi Noetic's F-BASIC 2.0 by Robert J. Tiess
"F-BASIC is a complete language and development system."
Gunship reviewed by Derek Perry MicroProse's aircraft combat.
Paperboy reviewed by R. J, Hicks Being a paperboy is easy?
Archipelagos reviewed by Miguel Mulet
Can you save the world??
Blood Money reviewed by Joe DiCara
A challenging action game from Psygnosis, Ltd.

Programming:

Leggo My LOGO by MIke Morrison
Logo's turtle teaches kids computer programming.
A CLI Beginner's Questions Answered by Mike Morrison
Those first questions new Amiga owners ask about CLI.
The Command Line by Rich Falconburg
Exploring commands in AmigaDOS.
Trees and Recursion by Forest W. Arnold
What are trees and recursion.
C Notes by Stephen Kemp
Two compressing data techniques.

Public Domain:

No Fishing! by Graham Kinsey
Great PD programs including slideshow program, ShoWiz.
PD Serendipity by Mike Morrison
A review of Escape From Jovi III and Mach 2 utility.

Columns:

New Products and Other Neat Stuff! by Elizabeth G. Fedorzyn
T.A.C.L., Celebrity Cookbook, Swords of Twilight, more
Video Schmideo by Barry Solomon
The 1989 Video & Graphics round-up.
Snapshot by Brad Andrews
Bug Bytes by John Steiner Bug fixes in WordPerfect's update.
Also some problems with DeluxePaint III.
Roomers by The Bandito A sneak peek at Apple's LCM, and the Amiga 3000 may be closer than you think.

Plus much more!

AC provides more Amiga information to its readers than any other resource. Your best Amiga value!

Subscribe Today!

Amazing COMPUTING™ For The Commodore AMIGA®
Your Original AMIGA® Monthly Resource

Index of Advertisers

Need more information?

Please use the Reader Service Card to contact those advertisers who have sparked your interest. Advertisers want to hear from you. This is the best way they have of determining the Amiga community's interests and needs. Take a moment and contact the companies with products you want to know more about. And, if you wish to contact an Advertiser directly, please tell them you saw their advertisement in Amazing Computing For The Commodore Amiga

Advertiser	Page	Reader Service Number
ACDA Corporation	89	104
Anivision	70	150
ACDA Corporation	62	104
AmiEXPO	9	115
Anivision	61	150
Central Coast Software	5	145
Computability	30	117
Fairbrother & SoeparMann	37	179
Joe's First Company, Inc.	11	178
KarmaSoft	79	124
Michigan Software	37	116
Micro Momentum, Inc.	77	188
Micro Momentum, Inc.	73	125
MicroMiga	20	182
MiGraph	13	138
MJ Systems	36	149
Natural Graphics	95	160
NewTek	CIV	140
One Byte Computers	71	135
Poor Person Software	85	127
Practical Solutions,Inc.	28	137
Pre'spect Technics, Inc.	93	165
Radical Eye Software	75	152
Safe Harbor Software & Peripherals	29	134
Software Plus	85	143
StraightLine Software, Inc.	38	166
Supra Corporation	7	168
The Bit Bucket Computer Store	60	139
The Grapevine Group	105	147
The Krueger Company	14	118
The Memory Location	68	107
The Memory Location	37	186
The Sterling Connection	85	157
Virtual Reality Laboratories, Inc.	69	121
Xetec	31	123

BECOME WORLD FAMOUS!
BE SEEN IN
ALL THE BEST PLACES!











Write For AC

Share your expertise with others and **get paid for it!**

Artists, Programmers, Hardware Hackers, Animators

People all over the world are waiting to read your tips, tricks and techniques.
Contact our Editorial Department today for details!

PiM Publications
Editorial Department
P.O. Box 869
Fall River, MA 02722-0869
(508) 678-4200

Amiga Circuits

Hardware Access To Your Amiga

by John Iovine

In the last few installments we learned to control the output and read the input of the Amiga parallel port. One limitation has been the minute amounts of current available to us from the port, just barely enough current to drive some of the miniature LED's. This month we will use those signal voltages at the port to control larger amounts of electrical power.

REAL WORLD ENVIRONMENT

The real world environment is where we live. Most computers do not impact in the real world. By impact, I mean they don't control anything beyond their own circuitry. The simple interfaces we will construct in this article will allow the computer to control various electrical devices in the real world.

By using the +5 volt TTL (Transistor-Transistor Logic) from any PB line on the parallel port, the computer will be able to control power flow (turning electrical power on or off) to various circuits or appliances. The circuits described can control either standard 115V AC (alternating current) electricity from your home or DC (direct current) electricity. Naturally, by controlling the electric current to a device, you are controlling the device.

INDUCTIVE AND RESISTIVE LOADS

Any device we are powering or controlling is called a load. Whatever the electrical device is it will fall into one of two main electrical categories, inductive or resistive loads.

It is pretty easy to distinguish an inductive load from a resistive load. An inductive device has coils or electrical wire windings in it as do motors, transformers, relays and solenoids. A resistive device has no coils or windings (as in incandescent lights, coffee makers, toasters, heaters).

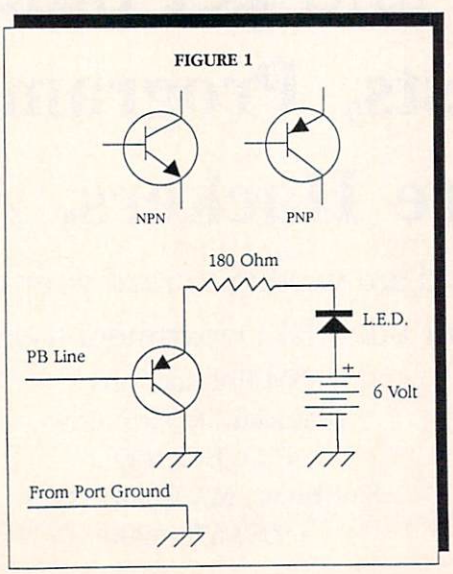
The distinction between loads is important because, when electric power is cut off to an inductive device, an electric pulse (called a transient) is generated. This electric pulse should be absorbed by a simple capacitor-diode combination in the circuit. Failure to do so may damage our circuit or computer. Fortunately, adding a capacitor-diode combination is very easy to do.

DC LOADS

Look at Figure One. This is a basic DC control circuit that uses a bi-polar transistor as a switch. This circuit is very useful. It can be used almost anywhere in a circuit that a switch can be used. As an example, using four of these transistor switches connected to four consecutive PB lines would allow you to control a stepper motor. Stepper motors are useful in robotics, positioning devices, printers, and disk drives.

Our drawing shows an PNP (positive, negative, positive) transistor. The base of the transistor is connected to the PB line. The base is also the "N" in PNP. When the base of an PNP transistor is at 0 volts (ground), current will flow through it. When we apply a binary "1" to the base (+5V), no current will flow. (The electrical polarity for the NPN (negative, positive,negative) transistor is opposite!) In our demo circuits this means that outputting a binary "0" turns the circuit on while a binary "1" shuts it off.

FIGURE 1



PNP VS. NPN TRANSISTORS

We are using PNP transistors in all the demo circuits. You may think that a NPN transistor would be a better choice since the transistor would turn on when you output a binary "1" on its PB line. The reason I didn't do this is that the current off any of the PB lines on the parallel port sources is a minute amount of current, just barely enough to turn a NPN transistor on. And we want the transistor to turn on or off completely. This makes the PNP transistor a better choice.

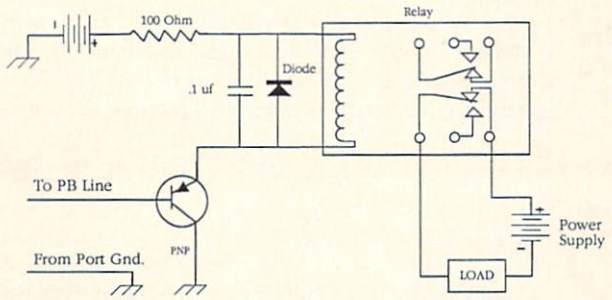
In addition, we have another advantage in using PNP transistors. When you power up your Amiga , even though the parallel port is configured as an input on power up, the PB lines are drawn up to +5 volts by pull up resistors, this effectively keeps any circuit you have connected in an off condition. This allows you time to load in AmigaBASIC and reconfigure the port.

AC OR DC LOADS

Examine Figure Two. This circuit can control AC or DC loads up to 120 volts at 2 amps. In this particular circuit we are not concerned as to whether the main load is resistive or inductive, it could be either. But notice the diode and capacitor connected across the relay. This is because the relay itself is an inductive load. The diode and capacitor are necessary to prevent damage to the transistor or computer. You can, if you wish, substitute an LED in place of the diode (make sure the polarity of LED is correct) and watch the LED flash every time the computer shuts off the circuit. This flash represents the voltage pulse being

Please Note:
This project may void your warranty and is offered for the enjoyment of the technically inclined. PiM Publications, Inc. is not responsible for any damages incurred while attempting this hardware project.

Figure 2



absorbed. The reason you can replace the diode with an LED is that an LED is a diode. (LED is an acronym for Light Emitting Diode.)

AC LOADS

The circuits shown in Figures Three and Four are exclusively for AC loads: resistive and inductive respectively. The six pin MOC 3010 chip is an opto-coupled triac available from Radio-Shack. This device senses when the PB line connected to it is outputting a binary "1" (+5V) with an internal LED. The internal LED triggers a photosensitive internal triac that in turn triggers the external triac, that powers our load.

APPLIANCE CONTROL

Although the circuits presented here are capable of controlling household appliances, this is NOT the way to control appliances in the home. There is a much more elegant way, using X-10 home power control modules. These modules are available from a number of retail outlets across the country. The X-10 power system uses your home electrical wiring system to transmit control codes to appliance modules that can turn appliances on, off, or to dim lights etc. Rather than these simple circuits that require their own wiring from the wall socket to the computer interface to the appliance itself. I have scheduled an X-10 home control system interface project in a few months, so if this is your application, hold off until then.

CIRCUIT CONSTRUCTION

We can build our transistor switch on A OR THE proto-typing bread board. This is a relatively simple matter, however, the amount of power available from pin 14 (+5V) on the parallel port is a mere 10 ma (milliamps). This forces us to supplement any circuit we interface to the port with its own power supply. For the demo circuit we are working with we can supplement the power using a 6V or 9V battery. You can connect any device (load) you'd like in the circuit, an LED or a small DC motor. If you use a motor remember to add the diode and capacitor. You should be able to control the devices using the basic commands we worked with in the last two installments.

With the line voltage AC circuits you must be much more careful. If you build these circuits please remember that the power available from your household electrical outlet is more than enough to reduce your computer to a cinder or give yourself a nasty (possibly fatal) shock.

I advise constructing the inductive load (fig. 4) circuit for any and all applications, as this circuit can be used with either types of loads. This will alleviate any potential problems in the future. I am, however, including the schematics for the resistive loads for comparison or as dedicated resistive load controllers.

You must not use the proto-typing board with the line voltage circuits because the board is not designed to handle the voltages and current. The components should be soldered to a PC board for safety.

TROUBLE SHOOTING

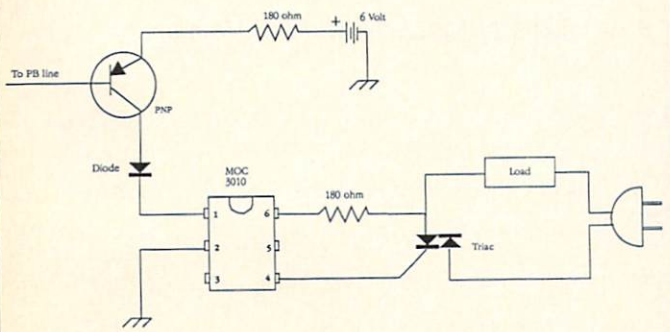
With the MOC 3010 chip, the internal diode that connects to the PNP transistor line is quite sensitive. The diode will burn out if you inadvertently reverse the polarity, even if it is just momentarily. Therefore, I added an additional diode to prevent that from happening. If you use an LED in place of the diode in this circuit it will act as a on indicator.

•AC•

Parts List

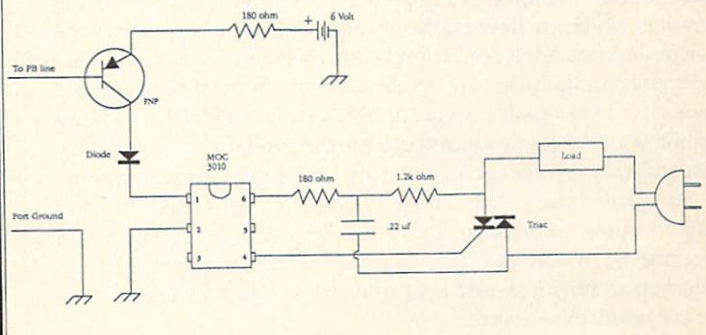
Description	Radio-Shack Part No.
180 Ohm resistor	271-014
1.2 K resistor	271-024
2.2 K resistor	271-027
Triac 6 Amp 200V	271-1001
.1 uF Capacitor	272-135
.22 uF Capacitor	272-1070
Opto-Coupler MOC3010	276-134
NPN Transistors	276-1617
Subminiature LED	276-026B
Relay	275-243

FIGURE 3



Resistive Load 120 VAC

FIGURE 4



A.C. Inductive Load 120 VAC

(Scanning, continued from page 85)

```
Vertical.Pixels = 400
```

The program has been written to automatically take care of all other details. However, be warned! The program will take twice as long to complete the fractal!

THIS MONTH'S EXAMPLE

Looking through this month's listing and identifying the sections which scale and scan the screen, a line which has not appeared in previous examples should become noticeable:

```
DEFDBL xmin, xmax, ymin, ymax, dx, dy, x, y, xk, i, j,
r
```

The need for this is simple. Remembering back to the second article, it was suggested to see if there was any limit to how much the Mandelbrot set could be magnified (by scaling xmin, xmax, ymin, and ymax for smaller and smaller ranges). You will discover that beyond a certain point the program does not operate anymore. It just seems to hang up, not generating anything. That limit is related to the mathematical accuracy of the computer. Try the following interactive example. Fire up AmigaBASIC, and enter into the OUTPUT window:

```
print 22/7
```

Answer:

```
3.142857
```

The number was stored in the computer to an accuracy of seven digits. The information beyond that was lost. Normally this does not present a problem because those lost digits are a very small fraction of the final answer. They contribute little to the normal types of calculations that most people expect computers to do. But sometimes more accuracy is really required.

Enter the following:

```
print 22#/7#
```

Answer:

```
3.142857142857143
```

Sixteen digit accuracy. That's called double precision arithmetic. The Amiga offers it for those rare occasions when extra accuracy is needed. In this month's example the screen has been scaled using numbers which are beyond the normal seven digit accuracy of the Amiga, and as a result double precision arithmetic is required. That is done in AmigaBASIC by either using the '#' symbol as the last character in the variable name, or by using the DEFDBL command (define double). It's much easier to use the DEFDBL command than it is to add a '#' to every occurrence of every variable in the program. If, at a later time, the screen needed to be scaled for less magnification, conversion back to single precision arithmetic can be done by modifying only one line in the program. Double precision arithmetic is used only when it is needed because it slows the computer down a bit.

As the numerical values for xmin, xmax, ymin, and ymax are entered the Amiga automatically adds a '#' character to the end of the line. That's the Amiga alerting the user to the need for storing these numbers in double precision variables. Great machine, that Amiga!

So what about TRUE BASIC? Well, it's numerical accuracy is already 14 digits, only two digits less than AmigaBASIC'S double precision. It doesn't need to use double precision arithmetic to generate this month's example. Enter the short program:

```
print using "#.############": 22/7
end
```

```
The result:      3.1428571428571
```

Note that the TRUE BASIC print statement displays numbers to 7 digit accuracy unless specifically asked to display more, as done above. However, numbers are always stored and calculations are always performed to an accuracy of 14 digits.

The time advantages achieved by other compiled BASIC's is achieved mostly by cutting back on the accuracy of the calculations performed. This accuracy becomes more important when greater magnification of the Mandelbrot set is wanted.

The next obvious question is what about TRUE BASIC's double precision arithmetic? Sorry, for now it is limited to 14 digits. TRUE BASIC handles the double precision question by providing a Mathematician's Toolkit for each machine. Unfortunately, the Math Toolkit for the Amiga has not been released yet. It has been released for the IBM, MAC, and ATARI-ST, and provides those users with the ability to perform extremely high precision calculations, to 100,000 digits! It also capable of handling complex number calculations. And the Mandelbrot set is actually based on the repetitive squaring of complex numbers, but that is another lesson.

This month's example represents a fractal that I designed myself. I scaled the screen using coordinates that were close (but not exactly equal) to some of the examples in the book "The Beauty of Fractals by Peitgen & Richter, Springer-Verlag 1986". I then made my own color decisions, thus producing my own artistic effects. More about this in a future article.

I have written this month's example using the same method of saving the picture that was developed in my last article in this series. Referring to that article, MERGE the 'SaveILBM' program from the Extras disk to the end of this month's listing. If GRABBIT is being used, or if a better method of saving fractals to disk is available, simply delete the related lines as directed by comments in the listing.

The 'Calculate' and 'Select.Color' subroutines are the only parts of the fractal generating program that have not yet been discussed. Next months lesson will begin explaining why these routines produce such complex pictures.

Listing One, AmigaBASIC Version

```
REM ************************************************************
REM *                  Sixteen Color Fractal
REM *
REM *                    by Paul Castonguay
REM *
REM ************************************************************

DEFDBL xmin, xmax, ymin, ymax, dx, dy, x, y, xk, i, j, r

DEF FNx(x) = INT(((x-xmin)+dx/2)/dx)
DEF FNy(y) = (Vertical.Pixels - 5) - INT(((y-ymin)+dy/2)/
dy)
```

```
xmin = -.74543701#
xmax = -.74542009#
ymin =  .11300268#
ymax =  .11301492#

Vertical.Pixels = 200

dx = (xmax-xmin)/631

IF Vertical.Pixels = 200 THEN
    dy = (ymax-ymin)/195
    SCREEN 1, 640, 200, 4, 2
    WINDOW 2, , (0,0)-(631,186), 0, 1
    Crunch = 800
ELSEIF Vertical.Pixels = 400 THEN
    dy = (ymax-ymin)/395
    SCREEN 1, 640, 400, 4, 4
    WINDOW 2, , (0,0)-(631,386), 0, 1
    Crunch = 1600
ELSE
    PRINT "Set Vertical.Pixels to 200 or 400"
    GOTO Done
END IF

GOSUB Choose.Color.Numbers
GOSUB Use.New.Colors

M = 4
CLS

'*****************************************************************
'
'       Delete next two lines if you're using GRABBIT
'
'*****************************************************************
LOCATE 23, 20
INPUT "Enter name to save fractal "; ILBMname$

LOCATE 10, 23
PRINT "... Fractal is being generated ..."

FOR j=ymin TO ymax+dy/2 STEP dy
  FOR i = xmin TO xmax+dx/2 STEP dx
    GOSUB Calculate
    GOSUB Select.Color
    PSET (FNx(i),FNy(j))
    IF INKEY$ = CHR$(138) THEN GOTO Done
  NEXT i
NEXT j

REM Wait for operator to press function key [F-10]
WHILE INKEY$ <> CHR$(138)
WEND

Done:

'*****************************************************************
'             Delete next line if you're using GRABBIT
'
'*****************************************************************
GOSUB Save.To.Disk

WINDOW CLOSE 2
SCREEN CLOSE 1

END


Calculate:
  x = 0
  y = 0
  k = 0
  r = 0
  WHILE r<=M AND k<Crunch
    xk = x*x - y*y + i
    y = 2*x*y + j
    x = xk
    k = k+1
    r = x*x + y*y
  WEND
RETURN

Select.Color:
  IF k = Crunch THEN
    COLOR 0
  ELSEIF k>=600 THEN
    COLOR 15
  ELSEIF k>=500 THEN
    COLOR 14
  ELSEIF k>=400 THEN
    COLOR 13
  ELSEIF k>=300 THEN
    COLOR 12
  ELSEIF k=169 OR k=167 OR k=165 THEN
    COLOR 11
  ELSEIF k=168 OR k=166 OR k=164 OR k=162 OR k=160 THEN
    COLOR 10
  ELSEIF k=163 OR k=161 OR k=159 OR k=157 THEN
    COLOR 9
  ELSEIF k=158 OR k=156 OR k=154 OR k=152 OR k=150 THEN
    COLOR 8
  ELSEIF k=155 OR k=153 OR k=151 OR k=149 OR k=147 THEN
    COLOR 7
  ELSEIF k=148 OR k=146 OR k=144 OR k=142 OR k=140 THEN
    COLOR 6
  ELSEIF k=145 OR k=143 OR k=141 OR k=139 OR k=137 THEN
    COLOR 5
  ELSEIF k=138 OR k=136 OR k=134 OR k=132 OR k=130 THEN
    COLOR 4
  ELSEIF k=135 OR k=133 OR k=131 OR k=129 THEN
    COLOR 3
  ELSEIF k=128 OR k=126 OR k=124 OR k=122 OR k=120 THEN
    COLOR 2
  ELSEIF k=127 OR k=125 OR k=123 OR k=121 THEN
    COLOR 1
  ELSE
    COLOR 0
  END IF
RETURN

Choose.Color.Numbers:
  LOCATE 10,20
  PRINT "... Please wait while I adjust colors ..."
  Reg.0.Red   =  2
  Reg.0.Green =  0
  Reg.0.Blue  =  2
  Reg.1.Red   =  11
  Reg.1.Green =  13
  Reg.1.Blue  =  15
  Reg.2.Red   =  0
  Reg.2.Green =  0
  Reg.2.Blue  =  0
  Reg.3.Red   =  9
  Reg.3.Green =  12
  Reg.3.Blue  =  14
  Reg.4.Red   =  1
  Reg.4.Green =  1
  Reg.4.Blue  =  1
  Reg.5.Red   =  8
  Reg.5.Green =  11
  Reg.5.Blue  =  13
  Reg.6.Red   =  2
  Reg.6.Green =  2
  Reg.6.Blue  =  2
  Reg.7.Red   =  7
  Reg.7.Green =  10
  Reg.7.Blue  =  13
  Reg.8.Red   =  3
  Reg.8.Green =  3
  Reg.8.Blue  =  3
  Reg.9.Red   =  6
  Reg.9.Green =  9
  Reg.9.Blue  =  12
  Reg.10.Red  =  4
  Reg.10.Green =  4
  Reg.10.Blue  =  4
  Reg.11.Red   =  6
  Reg.11.Green =  8
  Reg.11.Blue  =  11
  Reg.12.Red   =  0
```

```
    Reg.12.Green = 6
    Reg.12.Blue  = 10
    Reg.13.Red   = 12
    Reg.13.Green = 4
    Reg.13.Blue  = 0
    Reg.14.Red   = 14
    Reg.14.Green = 8
    Reg.14.Blue  = 0
    Reg.15.Red   = 12
    Reg.15.Green = 12
    Reg.15.Blue  = 0
RETURN

Use.New.Colors:
 PALETTE  0,  Reg.0.Red/16,  Reg.0.Green/16,  Reg.0.Blue/16
 PALETTE  1,  Reg.1.Red/16,  Reg.1.Green/16,  Reg.1.Blue/16
 PALETTE  2,  Reg.2.Red/16,  Reg.2.Green/16,  Reg.2.Blue/16
 PALETTE  3,  Reg.3.Red/16,  Reg.3.Green/16,  Reg.3.Blue/16
 PALETTE  4,  Reg.4.Red/16,  Reg.4.Green/16,  Reg.4.Blue/16
 PALETTE  5,  Reg.5.Red/16,  Reg.5.Green/16,  Reg.5.Blue/16
 PALETTE  6,  Reg.6.Red/16,  Reg.6.Green/16,  Reg.6.Blue/16
 PALETTE  7,  Reg.7.Red/16,  Reg.7.Green/16,  Reg.7.Blue/16
 PALETTE  8,  Reg.8.Red/16,  Reg.8.Green/16,  Reg.8.Blue/16
 PALETTE  9,  Reg.9.Red/16,  Reg.9.Green/16,  Reg.9.Blue/16
 PALETTE  10,  Reg.10.Red/16,  Reg.10.Green/16,  Reg.10.Blue/16
 PALETTE  11,  Reg.11.Red/16,  Reg.11.Green/16,  Reg.11.Blue/16
 PALETTE  12,  Reg.12.Red/16,  Reg.12.Green/16,  Reg.12.Blue/16
 PALETTE  13,  Reg.13.Red/16,  Reg.13.Green/16,  Reg.13.Blue/16
 PALETTE  14,  Reg.14.Red/16,  Reg.14.Green/16,  Reg.14.Blue/16
 PALETTE  15,  Reg.15.Red/16,  Reg.15.Green/16,  Reg.15.Blue/16
RETURN

REM*********************************************************
REM         delete from here to end if you are using GRABBIT
REM*********************************************************

Save.To.Disk:

    ccrtDir%   = 0
    ccrtStart% = 0
    ccrtEnd%   = 0
    ccrtSecs&  = 0
    ccrtMics&  = 0

    DIM bPlane&(5), cTabSave%(32)

    DECLARE FUNCTION xOpen&  LIBRARY
    DECLARE FUNCTION xRead&  LIBRARY
    DECLARE FUNCTION xWrite& LIBRARY
    DECLARE FUNCTION AllocMem&() LIBRARY

    LIBRARY "dos.library"
    LIBRARY "exec.library"
    LIBRARY "graphics.library"

    GOSUB SaveILBM

RETURN
```

Listing Two, True BASIC Version

```
REM*********************************************************
REM
REM              Add Extras:Demos/SaveILBM to end of listing
REM
REM                 using MERGE command from AmigaBASIC
REM
REM                 OUTPUT window  (see text, Article #3)
REM
REM*********************************************************
!*********************************************************
!                      Sixteen Color Fractal
!
!                        by Paul Castonguay
!
!*********************************************************

let Vertical_Pixels = 200

let xmin = -.74543701
let xmax = -.74542009
let ymin =  .11300268
let ymax =  .11301492
let dx = (xmax-xmin)/639

if Vertical_Pixels = 200 then
    set mode "HIGH16"
    let dy = (ymax-ymin)/199
    let Crunch = 800
else
    if Vertical_Pixels = 400 then
        set mode "LACEHIGH16"
        let dy = (ymax-ymin)/399
        let Crunch = 1600
    else
        print "Set Vertical_Pixels to 200 or 400"
        stop
    end if
end if

call Choose_Color_Numbers
call Use_New_Colors

set window xmin, xmax, ymin, ymax

let M = 4

set cursor 10,23
print "... Fractal is being generated ..."

for j=ymin to ymax+dy/2 step dy
    for i=xmin to xmax+dx/2 step dx
        call Calculate
        call Select_Color
        plot points: i,j
        call Test_F10
    next i
next j


! Wait for operator to press function key [F-10]
call Wait_Response

sub Calculate
        let x=0
        let y=0
        let k=0
        let xk=0
        let r=0
        do while (r<M and k<Crunch)
           let xk = x*x - y*y +i
           let y = 2*x*y + j
           let x = xk
           let k = k+1
           let r = x*x + y*y
        loop
end sub

sub Select_Color
    if k = Crunch then
        set color 0
    else
        select case k
                    case is > 600
                    set color 15
                case 500 to 599
                    set color 14
                case 400 to 499
                    set color 13
                 case 300 to 399
                    set color 12
                case 169, 167, 165
                    set color 11
                case 168, 166, 164, 162, 160
                    set color 10
                case 163, 161, 159, 157
                    set color 9
                 case 158, 156, 154, 152, 150
```

```
                set color 8
            case 155, 153, 151, 149, 147
                set color 7
            case 148, 146, 144, 142, 140
                set color 6
            case 145, 143, 141, 139, 137
                set color 5
            case 138, 136, 134, 132, 130
                set color 4
            case 135, 133, 131, 129
                set color 3
            case 128, 126, 124, 122, 120
                set color 2
            case 127, 125, 123, 121
                set color 1
            case else
                set color 0
        end select
    end if
end sub

sub Choose_Color_Numbers
    let Reg_0_Red    = 2
    let Reg_0_Green  = 0
    let Reg_0_Blue   = 2
    let Reg_1_Red    = 11
    let Reg_1_Green  = 13
    let Reg_1_Blue   = 15
    let Reg_2_Red    = 0
    let Reg_2_Green  = 0
    let Reg_2_Blue   = 0
    let Reg_3_Red    = 9
    let Reg_3_Green  = 12
    let Reg_3_Blue   = 14
    let Reg_4_Red    = 1
    let Reg_4_Green  = 1
    let Reg_4_Blue   = 1
    let Reg_5_Red    = 8
    let Reg_5_Green  = 11
    let Reg_5_Blue   = 13
    let Reg_6_Red    = 2
    let Reg_6_Green  = 2
    let Reg_6_Blue   = 2
    let Reg_7_Red    = 7
    let Reg_7_Green  = 10
    let Reg_7_Blue   = 13
    let Reg_8_Red    = 3
    let Reg_8_Green  = 3
     let Reg_8_Blue    = 3
    let Reg_9_Red    = 6
    let Reg_9_Green  = 9
    let Reg_9_Blue   = 12
    let Reg_10_Red   = 4
    let Reg_10_Green = 4
    let Reg_10_Blue  = 4
    let Reg_11_Red   = 6
    let Reg_11_Green = 8
    let Reg_11_Blue  = 11
    let Reg_12_Red   = 0
   let Reg_12_Green = 6
   let Reg_12_Blue  = 10
  let Reg_13_Red    = 12
   let Reg_13_Green = 4
    let Reg_13_Blue  = 0
    let Reg_14_Red   = 14
    let Reg_14_Green = 8
    let Reg_14_Blue  = 0
    let Reg_15_Red   = 12
    let Reg_15_Green = 12
    let Reg_15_Blue  = 0
end sub

sub  Use_New_Colors
     set color mix (0) Reg_0_Red/16, Reg_0_Green/16,
Reg_0_Blue/16
     set color mix (1) Reg_1_Red/16, Reg_1_Green/16,
Reg_1_Blue/16
    set color mix (2) Reg_2_Red/16, Reg_2_Green/16, Reg_2_Blue/
16
     set color mix (3) Reg_3_Red/16, Reg_3_Green/16,
Reg_3_Blue/16
     set color mix (4) Reg_4_Red/16, Reg_4_Green/16,
Reg_4_Blue/16
    set color mix (5) Reg_5_Red/16, Reg_5_Green/16, Reg_5_Blue/
16
    set color mix (6) Reg_6_Red/16, Reg_6_Green/16, Reg_6_Blue/
16
    set color mix (7) Reg_7_Red/16, Reg_7_Green/16, Reg_7_Blue/
16
    set color mix (8) Reg_8_Red/16, Reg_8_Green/16, Reg_8_Blue/
16
    set color mix (9) Reg_9_Red/16, Reg_9_Green/16, Reg_9_Blue/
16
    set color mix (10) Reg_10_Red/16, Reg_10_Green/16,
Reg_10_Blue/16
    set color mix (11) Reg_11_Red/16, Reg_11_Green/16,
Reg_11_Blue/16
    set color mix (12) Reg_12_Red/16, Reg_12_Green/16,
Reg_12_Blue/16
    set color mix (13) Reg_13_Red/16, Reg_13_Green/16,
Reg_13_Blue/16
    set color mix (14) Reg_14_Red/16, Reg_14_Green/16,
Reg_14_Blue/16
    set color mix (15) Reg_15_Red/16, Reg_15_Green/16,
Reg_15_Blue/16
end sub

sub Wait_Response
    do
        if key input then
            get key k
            if k=324 then exit do
        end if
     loop
     clear
    set cursor 10,17
    print "... Press left mouse button to clear screen ..."
end sub

sub Test_F10
    if key input then
        get key press_f10
        if press_f10 = 324 then
            clear
            set cursor 10,17
            set color 1
            print "... Press left mouse button to clear screen
..."
            stop
        end if
    end if
end sub

end
```


•AC•

AC's GUIDE TO THE COMMODORE AMIGA

What's that **program** called?
Where can I get some more
information about that item?
Doesn't **anybody** make something
that does that?
These questions and more answered
in AC's **GUIDE**
Over 2200 products listed.
On sale at your Amazing Dealer!

C Notes

from the C Group

Recursive Functions

by Stephen Kemp

Recursion (a function calling itself) is possible in the C language. It isn't often that you will require a recursive function, but occasionally such a function can solve the most difficult of tasks. Recursion simply means that a function calls itself — either directly or indirectly — by calling other functions that eventually call the first function.

As a demonstration of how recursion works, examine the function in Listing One. This is a rather contrived and simple example, but it demonstrates how recursion works. The function "backwards" accepts a string of characters as a parameter. It takes the string and prints it in the reverse order.

Listing One

```
/*Function backwards accepts a string and calls itself
recursively to print the string in reverse*/

backwards( string )
char *string;
{
     if (*string == '\0') /*if at the end of the string*/
        return;                    /*nothing to do here*/
     backwards( string+1 )/*recursive call with shorter
                                   string*/
     print("%c",*string); /*print the character*/
}
```

Examining the function, you can see that it first checks to see if the end of the string has been reached. If the first character is a null, then it simply returns to the caller. If it is not at the end of the string, this function shortens the string by the first character and then calls itself with the smaller string. Upon returning from itself, the function knows that it is time to print the original character that was kept.

Playing computer, let's look at how an example string would progress through this function. If you started with the string "How are you?", Table One represents the calls and output.

Although the table was built sequentially from the first call to the last, the characters actually print out in the reverse order because each function will only print a single character after the function returns. Incidentally, this function does not print its own carriage return or line feed. If you include this function in a sample program, you may have to print a CR/LF before you can see the output.

As explained, the example in Listing One is not a real practical example of how to use recursion. In fact, this function could be written much easier without it. Additionally, recursive functions should use prudence. This is due to the fact that each call to a function uses stack space. The stack is an area of memory reserved in your programs for remembering a calling function's address and local variables defined within functions. A certain amount of space is provided for the stack when your program begins, and although most versions of C will allow you to control that amount, you have to "live" within your choice. If your program uses up the allocated stack, then it will 1) fail with a stack overflow, or 2) start doing bizarre things and fail even worse.

Generally, there are three questions you should answer before deciding that you need a recursive function. First, does the function require a "loop" to perform its operation? Second, will the loop's operation depend upon a variable, or variables, determined by an earlier pass through the loop? Finally, can those intermediate values be determined without completing the function?

Table One

Received	Recursive Call	Print On Return
"Say what?"	"ay what?"	"S"
"ay what?"	"y what?"	"a"
"y what?"	" what?"	"y"
" what?"	"what?"	" "
"what?"	"hat?"	"w"
"hat?"	"at?"	"h"
"at?"	"t?"	"a"
"t?"	"?"	"t"
"?"	\0	"?"
\0		

Better examples of functions that might use recursion efficiently are parser and sorting functions. A parser is a function that takes its input and reduces it down to smaller and smaller components which can then be evaluated easier. Compilers use parsers to evaluate statements. Some statements in C, such as if, for, do, etc., can be composed of multiple statements. Evaluating one of these multi-line statements might require recursion to evaluate the individual statements contained within it. Now, I don't pretend that a compiler is as simple as that example, but I hope it helps you understand a little about how a parser works.

Now, not all sorting functions will have to be recursive. If you have been in programming very long, then you probably know that there are about as many different sorting algorithms as there are different languages. Listing Two contains a program that uses a recursive sorting function. This sort method is a version of "quick sort". (Note: Quick sort is a relative term. The actual speed of any sort depends upon the data being sorted and the relative order before the sort begins.)

Listing Two

```
/* QSORT.C is a program that takes an array of unsigned short
integers and quick sorts the elements using a function that
does recursion. */

#define MAXARR 15
short  arrange();

main()
{
    unsigned short ndx;                 /*index for array
                                         subscript*/
    unsigned short array[MAXARR];       /*array of elements*/
    unsigned short orig[MAXARR];        /*original for print*/

    srand(99);                          /*random number seed*/
    for(ndx = 0; ndx < MAXARR;ndx++){   /*initialize arrays*/
         array[ndx] = rand();           /*random fill*/
         orig[ndx] = array[ndx];        /*make a copy*/
    }

    doqsort(array,0,MAXARR-1);          /*sort array*/

    printf("Original   Sorted\r\n");  /*heading*/
    for(ndx = 0; ndx < MAXARR; ndx++) /*loop through arrays*/
         printf(" %6.6u     %6.6u\r\n",orig[ndx],array[ndx]);

}

/*this function accepts the array of unsigned short integers
and a starting and ending index.  It then sorts the values
through recursive calls to itself*/

doqsort(array, start, end)
unsigned short array[];
unsigned short start,end;
{
    short rval;

    if (start < end){                   /*if elements remain*/
         rval = arrange(array,start,end); /*arrange elements*/
         if (rval)                        /*if not first
                                            element*/
              doqsort(array, start, rval-1); /*sort new end*/
         doqsort(array, rval+1, end);       /*sort new
                                            begenning*/
    }
}

/* arrange from start to end */

short arrange(array, start, end)
unsigned short array[];
unsigned short start,end;
{
    unsigned short hold;                    /*holding value*/

    do{
         while( end > start && array[end] >= array[start])
                                           /*greater*/
              end--;                       /*reduce end*/

         if (end != start){                /*if not on start*/
              hold = array[start];         /*hold value*/
              array[start] = array[end];   /*swap values*/
              array[end] = hold;           /*swap values*/
              while( end > start && array[start] <= array[end])
/* less*/
                   start++;                /*increase start*/

              if (end != start){           /*not at end*/
                   hold = array[start];    /*hold value*/
                     array[start] = array[end]; /*swap
values*/
                   array[end] = hold;      /*swap values*/
              }
         }
    }while( start != end);                /*continue until equal*/

    return(start);                        /*return final
start*/
}
```

The quick sort function splits the elements into two parts, if it finds something out of order. Next, each part is sorted by making the recursive call to the sorting function. This continues until nothing remains out of order.

The arrange function simply begins with the last element passed and finds the element closest to the end of the list that should swap with the first element. If it finds one, then it swaps the elements. After that, it starts trying to find a new element from the beginning of the list to exchange the new ending position. If it finds one, then they are also swapped. This continues until the starting position equals the ending position. When it has been completed, the current starting position is returned and indicates a new sort position. If 0 is returned, then everything is in order.

With a few modifications, you can add this quick sorting capability to any of your programs. (MANX has a quick sort function in its library.) Primarily, you may have to alter the parameter that is being sorted to some other type, and you may have to change the comparison of elements to a function that returns a value less than, equal to, or greater than zero depending upon whether the sort should be ascending or descending. Finally, depending upon the variable type being sorted, it may be easier to use a function to swap the elements rather than doing it in line twice.

Now that you know recursion is permissable, you may discover just where it can help you in your programs. Just remember to keep in mind that there is a danger of doing recursion unwisely. But with a little experimenting, you will know exactly when and where to use your new knowledge.

•AC•

C= ™ • JUST RELEASED • AMIGA™

AMIGA™ UPGRADE #1 New 1 MEGABYTE, "FATTER Agnus (#8372) Hi-Res chip allows users more "chip" memory for use in GRAPHICS, MUSIC OR VIDEO. This "plug in" upgrade is an absolute must for present Amiga owners. Price is $109.50 including simple step by step, 10 minute installation instructons.

AMIGA™ UPGRADE #2 A501/801. 512K/clock RAM Board upgrade for Amiga 500. (Plugs directly into trapdoor.) Super price of $114.95 including instructions.

AMIGA™ UPGRADE #3 New . . V1.3 KICKSTART ROM upgrade-$27.95 including instructions.

AMIGA™ UPGRADE #4 Upgrade chips for all Amiga memories: 68020/33- $139.50, 68030-CALL, 68881/12-$69.95, 68881/16-$76.95, 68881/20-$104.95, 68882/16-$105.95, Other Speeds including Drams available at super prices.

WE STOCK ALL AMIGA (AND COMMODORE) CHIPS AT SUPER LOW PRICES

OUR 9th YEAR

• SEND FOR CATALOG OF EXCLUSIVE PRODUCTS •

THE GRAPEVINE GROUP, INC.
35 CHARLOTTE DRIVE
WESLEY HILLS, NY 10977

1-800-292-7445
Prices subject to change

(914) 354-4448
FAX (914) 354-6696
Dealer pricing available

• We ship worldwide •

VISA MasterCard

Circle 147 on Reader Service card.



AC Disks

Source code, executable, and IFF pictures (when available) included for all articles printed in *Amazing Computing*.



This disk contains the source and executable code relating to articles in AC V3.8 and AC V3.9.

Gels In MultiForth Parts I & II: Learn how to use Gels in MultiForth. Author: John Bushakra

FFP & IEEE: An Example of using FFP & IEEE math routines in Modula-2. Author: Steve Faiwiszewski

CAI: A complete Computer Aided Instruction program with editor written in AmigaBASIC. Author: Paul Castonguay

Tumblin' Tots: A complete game written in Assembly language. Save the falling babies in this game. Author: David Ashley

VGad: A gadget editor that allows you to easily create gadgets. The program then generates C code that you can use in your own programs. Author: Stephen Vermeulen

MenuEd: A menu editor that allows you to easily create menus. The program then generates C code.that you can use in your own programs. Author David Pehrson

Bspread: A powerful spread sheet.program written in AmigaBASIC. Author Brian Cately



This disk contains the source and executable code relating to articles in AC V4.3 and AC V4.4.

Fractals Part I: An introduction to the basics of fractals with examples in AmigaBASIC, True BASIC, and C. Author: Paul Castonguay

Shared Libraries: C source and executable code that shows the use of shared libraries. Author: John Baez

MultiSort: Sorting and intertask communication in Modula-2. Author: Steve Faiwiszewski

Double Playfield: Shows how to use dual playfields in AmigaBASIC. Author: Robert D'Asto

'881 Math Part I: Programming the 68881 math coprocessor chip in C Author: Read Predmore

Args: Passing arguments to an AmigaBASIC program from the CLI. Author: Brian Zupke



This disk contains the source and executable code relating to articles in AC V4.5 and AC V4.6.

Digitized Sound: Using the Audio.device to play digitized sounds in Modula-2. Author: Len A. White

'881 Math Part II: Part II of programming the 68881 math coprocessor chip using a fractal sample. Author: Read Predmore

At Your Request: Using the system-supplied requestors from AmigaBASIC. Author: John F. Weiderhirn

Insta Sound: Tapping the Amiga's sound from AmigaBASIC using the Wave command. Author: Greg Stringfellow

MIDI Out: A MIDI program that you can expand upon. Written in C. Author: Br. Seraphim Winslow

Diskless Compiler: Setting up a compiler environment that doesn't need floppies. Author: Chuck Raudonis



This disk contains the source and executable code relating to articles in AC V4.7 and AC V4.8.

Fractals Part II: Part II on fractals and graphics on the Amiga in AmigaBASIC and True BASIC. Author: Paul Castonguay

Analog Joysticks: The code for using analog joysticks on the Amiga. Written in C. Author: David Kinzer

C Notes: A small program to search a file for a specific string in C. Author: Steven Kemp

Better String Gadgets: How to tap the power of string gadgets in C. Author: John Bushakra

On Your Alert: Using the system's alerts from AmigaBASIC. Author: John F. Wiederhirn

Batch Files: Executing batch files from AmigaBASIC. Author: Mark Aydellotte

C Notes: The beginning of a utility program in C. Author: Steven Kemp



This disk contains the source and executable code relating to articles in AC V4.9.

Memory Squares: Test your memory with this AmigaBASIC game. Author: Mike Morrison

High Octane Colors: Use dithering in AmigaBASIC to get the appearance of many more colors. Author: Robert D'Asto

Cell Animation: Using cell animation in Modula-2. Author: Nicholas Cirasella

Improving Graphics: Improve the way your program looks no matter what screen it opens on. In C. Authro: Richard Martin

Gels in Multi-Forth-Part 3: The third and final part on using Gels in Forth. Author: John Bushakra

C Notes V4.9: Look at a simple utility program in C. Author: Steven Kemp

1D_Cells: A program that simulates a one-dimensional cellular automata. Author:Russell Wallace

Colourscope: A shareware program that shows different graphic designs. Author: Russell Wallace

ShowILBM: A program that displays lo-res, hi-res, interlace and HAM IFF pictures. Author: Russell Wallace

Labyrinth_II: Roll playing text adventure game. Author: Russell Wallace

Most: Text file reader that will display one or more files. The program will automatically format the text for you. Author: Russell Wallace

Terminator: A virus protection program. Author: Russell Wallace



This disk contains the source and executable code relating to articles in AC V4.10 & AC V4.11.

Typing Tutor: A program written in AmigaBASIC that will help you improvr your typing. Author: Mike Morrison

Glat's Gadgets: Using gadgets in Assembly language. Author: Geff Glatt

Function Evaluator: A program that accepts mathamatical functions and evaluates them. Written in C. Author: Randy Finch

Fractals: Part III: AmigaBASIC code that shows you how to save/load pictures to disk. Author: Paul Castonguay

More Requestors: Using system calls in AmigaBASIC to build requestors. Author: John Wiederhirn

Multi-Forth: Implementing the ARP library from Forth. Author: Lonnie A. Watson

Search Utility: A file search utility written in C. Author: Steven Kemp

Fast Pics: Re-writing the pixel drawing routine in assembly language for speed: Author:Scott Steinman

64 Colors: Using extra-half-brite mode in AmigaBASIC. Author: Bryan Catley

Fast Fractals: A fast fractal program written in C with assembly language subroutines. Author: Hugo M.H.Lyppens

Multitasking in Fortran: All the hard work is done here so you can multitask in Fortran. Author: Jim Locker



This disk contains the source and executable code relating to articles in AC V4.11 & AC V4.12.

Arexx Part II: Inofrmation on how to set up your own ARexx programs with examples. Author:Steve Gilmor.

Leggo My LOGO: A Logo program that generates a Christmas tree with decorations. Author: Mike Morrison.

Trees and Recursion: An introduction to binary trees and how to use recursion. Written in C. Author: Forest Arnold.

C Notes: A look at two data compressing techniques in C. Author: Stephen Kemp.

Animation? BASICally: Using cell animation with AmigaBASIC. Author: Mike Morrison

Menu Builder: A utility to help build menus in your own programs. Written in C. Author: Tony Preston.

Dual Demo: How to use dual playfields to make your own arcade games. Written in C. Author: Thomas Eshelman.

Scanning the Screen: Part four in the fractals series. This article covers drawing to the screen. In AmigaBASIC and TrueBasic. Author: Paul Castonguay.

C Notes: Recursive functions in C. Author: Stephen Kemp.

For PDS orders, please use form on page CIII
Visa and MasterCard is available on orders of $20.00 or more.

The AMICUS

Public Domain Software Library

This software is collected from user groups and electronic bulletin boards around the nation. Each Amicus disk is nearly full, and is fully accessible from the Workbench. If source code is provided for any program, then the executable version is also present. This means that you don't need the C compiler to run these programs. An exception is granted for those programs only of use to people who own a C compiler.

Note: Each description line below may include something like 'S-O-E-D', which stands for 'source, object file, executable and documentation'. Any combination of these letters indicates what forms of the program are present. Basic programs are presented entirely in source code format.

AMICUS Disk 1
ABasic programs: Graphics
3DSolids 3d solids modeling prog. w/sample data files
Blocks draws blocks
Cubes draws cubes
Durer draws pictures in the style of Durer
FScape draws fractal landscapes
Hidden 3D drawing program, w/ hidden line removal
JPad simple paint program
Optical draw several optical illusions
PaintBox simple paint program
Shuttle draws the Shuttle in 3d wireframe
SpaceArt graphics demo
Speaker speech utility
Sphere draws spheres
Spiral draws color spirals
ThreeDee 3d function plots
Topography artificial topography
Wheels draws circle graphics
Xenos draws fractal planet landscapes
ABasic programs: Tools
AddressBook simple database program for addresses
CardFile simple card file database program
Demo multiwindow demo
KeyCodes shows keycodes for a key you press
Menu run many ABasic programs from a menu
MoreColors way to get more colors on the screen at once, using aliasing
shapes simple color shape designer SpeakIt speech and narrator demo
ABasic programs: Games
BrickOut classic computer brick wall game
Othello also known as 'go'
Saucer simple shoot-em-up game
Spelling simple talking spelling game
ToyBox selectable graphics demo
ABasic programs: Sounds
Entertainer plays that tune
HAL9000 pretends it's a real computer
Police simple police siren sound
SugarPlum plays "The Dance of the Sugarplum Fairies"
C programs:
ATerm simple terminal program, S-E
cc aid to compiling with Lattice C
decvnt opposite of CONVERT for cross developers
Dotty source code to the 'dotty window demo
echox unix-style filename expansion, partial S,O-D
fasterfp explains use of fast-floating point math
FixDate fixes future dates on all files on a disk, S-E
freedraw simple Workbench drawing prog.,S-E
GfxMem graphic memory usage indicator, S-E
Grep searches for a given string in a file with docs.
ham shows off the hold-and-modify method of color generation
IBM2Amiga fast parallel cable transfers between an IBM and an Amiga
Mandel Mandelbrot set program, S-E
moire patterned graphic demo, S-E
objfix makes Lattice C object file symbols visible to Wack, S-E
quick quick sort strings routine
raw example sample window I/O
setface turns on interface mode, S-E
sparks qix-type graphic demo, S-E
Other executable programs:
SpeechToy speech demonstration
WhichFont displays all available fonts
Texts:
68020 describes 68020 speedup board from CSA
Aliases explains uses of the ASSIGN command
Bugs known bug list in Lattice C 3.02
CLICard reference card for AmigaDOS CLI
CLICommands guide to using the CLI
Commands shorter guide to AmigaDOS CLI commands
EdCommands guide to the ED editor
Filenames AmigaDOS filename wildcard conventions
HalfBright explains rare graphics chips that can do more colors
ModemPins description of the serial port pinout
RAMdisks tips on setting up your RAM: disk
ROMWack tips on using ROMWack
Sounds explanation of Instrument demo sound file format
Speed refutation of Amiga's CPU and custom chip speed
WackCmds tips on using Wack

AMICUS Disk 2
C programs:
alib AmigaDOS object library manager , S-E
ar text file archive program, S-E
fixobj auto-chops executable files
shell simple CLI shell, S-E
sq, usq file compression programs, S-E
YachtC a familiar game, S-E
Make a simple 'make' programming utility, S-E
Emacs an early version of the Amiga text editor, S-E-D
Assembler programs:
bsearch.asm binary search code
qsort.asm Unix compatible qsort() function, source and C test program
setjmp.asm setjmp() code for Lattice 3.02
SVprintf Unix system V compatible printf()
trees.o Unix compatible tree() function, O-D
(This disk formerly had IFF specification files and examples. Since this spec is constantly updated, the IFF spec files have been moved to their own disk in the AMICUS collection.)
John Draper Amiga Tutorials:
Animate describes animation algorithms
Gadgets tutorial on gadgets
Menus learn about Intuition menus

AMICUS Disk 3
C programs:
Xref a C cross-reference gen., S-E
6bitcolor extra-half-bright chip gfx demo, S-E
Chop truncate (chop) files down to size, S-E
Cleanup removes strange characters from text files
CR2LF converts carriage returns to line feeds in Amiga files, S-E
Error adds compile errors to a C file, S
Hello window ex. from the RKM, S
Kermit generic Kermit implementation, flakey, no terminal mode, S-E
Scales sound demo plays scales, S-E
SkewB Rubik cube demo in hi-res colors, S-E
AmigaBasicProgs(dir)
Automata cellular automata simulation
CrazyEights card game
Graph function graphing programs
WitchingHour a game
ABasiC programs:
Casino games of poker, blackjack, dice, and craps
Gomoku also known as 'othello'
Sabotage sort of an adventure game
Executable programs:
Disassem a 68000 disassembler, E-D
DpSlide shows a given set of IFF pictures, E-D
Arrange a text formatting program, E-D
Assembler programs:
Argoterm terminal program with speech and Xmodem, S-E

AMICUS Disk 4 Files from the original Amiga Technical BBS

Note that some of these files are old, and refer to older versions of the operating system. These files came from the Sun system that served as Amiga technical support HQ for most of 1985. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.

Complete and nearly up-to-date C source to 'image.ed', an early version of the Icon Editor. This is a little flaky, but compiles and runs.

An Intuition demo, in full C source, including files: demomenu.c, demomenu2.c, demoreq.c, getascii.c, idemo.c, idemo.guide, idemo.make, idemoall.h, nodos.c, and txwrite.c
addmem.c add external memory to the system
bobtest.c example of BOB use
consoleIO.c console IO example
creaport.c create and delete ports
creastdi.c create standard I/O requests
creatask.c creating task examples
diskio.c example of track read and write
dotty.c source to the 'dotty window' demo
dualplay.c dual playfield example
flood.c flood fill example
freemap.c old version of 'freemap'
geltools.c tools for VSprites and BOBs
gfxmem.c graphic memory usage indicator
hello.c window example from RKM
inputdev.c adding an input handler to the input stream
joystik.c reading the joystick
keybd.c direct keyboard reading
layertes.c layers examples
mousport.c test mouse port
ownlib.c,
ownlib.asm example of making your own library with Lattice
paratest.c tests parallel port commands
seritest.c tests serial port commands
serisamp.c example of serial port use
printr.c sample printer interface code
prtbase.h printer device definitions
regintes.c region test program
setlace.c source to interlace on/off program
setparallel.c set the attributes of the parallel port
SetSerial.c set the attributes (parity, data bits) of the serial port
singplay.c single playfield example
speechtoy.c source to narrator and phonetics demo
timedely.c simple timer demo
timer.c exec support timer functions
timrstuf.c more exec support timer functions
WhichFont.c loads and displays all available system fonts
process.i and prtbase.i assmebler include files:
autorqstr.txt warnings of deadlocks with autorequesters
consoleIO.txt copy of the RKM console I/O chapter
diskfont.txt warning of disk font loading bug
fullfunc.txt list of #defines, macros, functions
inputdev.txt preliminary copy of the input device chapter
License information on Workbench distribution license
printer pre-release copy of the chapter on printer drivers, from RKM 1.1
v11fd.txt 'diff' of .fd file changes from version 1.0 to 1.1 v28v1.diff 'diff' of include file changes from version 28 to 1.0

AMICUS Disk 5 Files from the Amiga Link / Amiga Information Network

Note that some of these files are old, and refer to older versions of the operating system. These files are from Amiga Link. For a time, Commodore supported Amiga Link, a GEnie, AIN, for online developer technical support. It was only up and running for several weeks. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.
A demo of Intuition menus called 'menudemo', in C source
whereis.c find a file searching all subdirectories
bobtest.c BOB programming example
sweep.c sound synthesis example
Assembler files:
mydev.asm sample device driver
mylib.asm sample library example
mylib.i
mydev.i
asmsupp.i
macros.i assembler include files
Texts:
amigatricks tips on CLI commands
extdisk external disk specification
gameport game port spec
parallel parallel port spec
serial serial port spec
v1.1update list of new features in version 1.1
v1.1h.txt 'diff' of include file changes from version 1.0 to 1.1
Files for building your own printer drivers, including despecial.c, epsondata.c, init.asm, printer.c, printer.link, printertag.asm, render.c, and wait.asm. This disk does contain a number of files describing the IFF specification. These are not the latest and greatest files, but remain here for historical purposes. They include text files and C source examples. The latest IFF spec is elsewhere in this library.

AMICUS Disk 6 IFF Pictures

This disk includes the DPSlide program, which can view a given series of IFF pictures, and the 'showpic' program, which can view each file at the click of an icon. The pictures include a screen from ArticFox, a Degas dancer, the guys at Electronic Arts, a gorilla, horses, King Tut, a lighthouse, a screen from Marble Madness, the Bugs Bunny Martian, a still from an old movie, the Dire Straits moving company, a screen from Pinball Contruction Set, a TV newcaster, the PaintCan, a world map, a Porsche, a shuttle mission patch, a tyrannosaurus rex, a planet view, a VISA card, and a ten-speed.

AMICUS Disk 7 DigiView HAM demo picture disk

This disk has pictures from the DigiView hold-and-modify video digitizer. It includes the ladies with pencils and lollypops, the young girl, the bulldozer, the horse and buggy, the Byte cover, the dictionary page, the robot and Robert. This includes a program to view each picture separately, and all together as separate, sidable screens. The 'seeibm' program, to turn any screen into an IFF picture.

AMICUS Disk 8
C programs:
Browse view text files on a disk, using menus S-E-D
Crunch removes comments and white space from C files, S-E
IconExec EXECUTE a series of commands from Workbench S-E
PDScreen Dump dumps Rastport of highest screen to printer
SetAlternate sets a second image for an icon, when clicked once S-E
SetWindow makes windows for a CLI program to run under Workbench S-E
SmallClock a small digital clockin a window menu bar
Scrimper the screen printer in the fourth AC S-E
Amiga Basic Programs:
(**Note**: Many of these programs are present on AMICUS Disk 1. Several of these were converted to Amiga Basic, & included here.)
AddressBook a simple address book database
Ball draws a ball
Cload program to convert Compuserve hex files to binary, S-D
Clue the game, Intuition driven
ColorArt art drawing program
DeluxeDraw the drawing program in the 3rd AC, S-D
Eliza conversational computer psychologist
Othello the game, as known as 'go'
RatMaze 3D ratmaze game
ROR boggling graphics demo
Shuttle draws 3D pictures of the space shuttle
Spelling simple spelling program
YoYo wierd zero-gravity yo-yo demo, tracks yo-yo to the mouse
Executable programs:
3Dcube Modula-2 demo of a rotating cube
AltIcon sets a second icon image, displayed when the icon is clicked
AmigaSpell a slow but simple spell checker, E-D
arc the ARC file compression prog must for telecom, E-D
Bertrand graphics demo
disksalvage prog. to rescue trashed disks, E-D
KwikCopy a quick but nasty disk copy program: ignores errors, E-D
LibDir lists hunks in an object file E-D
SaveILBM saves any screen as IFF pic.E-D ?? ScreenDump shareware screen dump prog, E only
StarTerm version 2.0, term program, XmodemE-D
Texts:
LatticeMain tips on fixing _main.c in Lattice
GDiskDrive make your own 5 1/4 drive
GuruMed explains the Guru numbers
Lat3.03bugs bug list of Lattice C version 3.03
MForgeRev user's view of the MicroForge HD
PrintSpooler EXECUTE-based print spool prog.
.BMAP files:
These are the necessary links between Amiga Basic and the system libraries. To take advantage of the Amiga's capabilities in Basic, you need these files. BMAPs are included for 'clist', 'console', 'diskfont', 'exec', 'icon', 'intuition', 'layers', 'mathffp', mathieeedoubas', 'mathieees-ingbas', 'mathtrans', 'potgo', 'timer' and 'translator'.

AMICUS Disk 9
Amiga Basic Programs:
FlightSim simple flight simulator program
HuePalette explains Hue, Saturation, & Intensity
Requester ex. of requesters from Amiga Basic
ScrollDemo demonstrates scrolling capabilities
Synthesizer sound program
WorldMap draws a map of the world
Executable programs:
Boing! latest Boing! demo,with selectable speed,E
Brush2C converts an IFF brush to C data instructions, initialization code, E
Brush2Icon converts IFF brush to an icon, E
Dazzle graphics demo, tracks to mouse, E
DeciGEL assembler program for stopping 68010 errors, S-E-D
Klock menu-bar clock and date display, E
life the game of life, E
TimeSet Intuition-based way to set the time & date
EMEmacs another Emacs, more oriented to word processing, S-E-D
MyCLI a CLI shell, works without the Workbench, S-E-D
Texts:
FnctnKeys read function keys from Amiga Basic
HackerSIn explains how to win the game 'hacker'
Ist68010 guide to installing a 68010 in your Amiga
Boing! latest Boing! demo,with selectablespeed, E
Brush2C converts an IFF brush to C data instructions, initialization code, E
Brush2Icon converts IFF brush to an icon, E
Dazzle graphics demo, tracks to mouse, E
DeciGEL assembler program for stopping 68010 errors, S-E-D
Klock menu-bar clock and date display, E
life the game of life, E
TimeSet Intuition-based way to set the time & date
EMEmacs another Emacs, more oriented to word processing, S-E-D
MyCLI a CLI shell, works without the Workbench, S-E-D
Texts:
FnctnKeys read function keys from Amiga Basic
HackerSIn explains how to win the game 'hacker'
Ist68010 guide to installing a 68010 in your Amiga
Boing! latest Boing! demo,with selectablespeed, E
Brush2C converts an IFF brush to C data instructions, initialization code, E
Brush2Icon converts IFF brush to an icon, E
Dazzle graphics demo, tracks to mouse, E
DeciGEL assembler program for stopping 68010 errors, S-E-D
Klock menu-bar clock and date display, E
life the game of life, E
TimeSet Intuition-based way to set the time & date
EMEmacs another Emacs, more oriented to word processing, S-E-D
MyCLI a CLI shell, works without the Workbench, S-E-D
Texts:
FnctnKeys explains how to read function keys from Amiga Basic
HackerSIn explains how to win the game 'hacker'
Ist68010 guide to installing a 68010 in your Amiga
PrinterTip sending escape sequences to your printer
StartupTip tips on setting up your startup-sequence file
XfrmrReview list of Transformer programs that work
Printer Drivers:
Printer drivers for the Canon PJ-1080A, the C Itoh Prowriter, an improved Epson driver that eliminates streaking, the Epson LQ-800, the Gemini Star-10, the NEC 8025A, the Okidata ML-92, the Panasonic KX-P10xx family, and the Smith-Corona D300, with a document describing the installation process.

AMICUS Disk 10 Instrument sound demos

This is an icon-driven demo, circulated to many dealers. It includes the sounds of an acoustic guitar, an alarm, a banjo, a bass guitar, a boink, a calliope, a car horn, claves, water drip, electric guitar, a flute, a harp arpegio, a kickdrum, a marimba, a organ minor chord, people talking, pigs, a pipe organ, a Rhodes piano, a saxophone, a sitar, a snare drum, a steel drum, bells, a vibrophone, a violin, a wailing guitar, a horse whinny, and a whistle.

AMICUS Disk 11
C programs
dirutil Intuition-based, CLI replacement manager S-E
cpri shows and adjusts priority of CLI processes, S-E
ps shows info on CLI processes, S-E
vidtex displays Compuserve RLE pics, S-E
AmigaBasic programs
pointered pointer and sprite editor program
optimize optimization ex ample from AC article
calendar large, animated calendar, diary and date book program
amortize loan amortizations
brushtoBOB converts small IFF brushes to AmigaBasic BOB OBJECTS
grids draw and play waveforms
hilbert draws Hilbert curves
madlib mad lib story generator
mailtalk talking mailing list program
meadows3D 3D graphics program, from A C™ article
mousetrack mouse tracking example in hires mode
slot slot machine game
tictactoe the game
switch pachinko-like game
weird makes strange sounds
Executable programs
cp unix-like copy command, E
cls screen clear, S-E
diff unix-like stream editor uses 'diff' output to fix files
pm chart recorder performances indicator
Assembler programs
cls screen clear and CLI arguments example
Modula-2
trails moving-worm graphics demo
caseconvert converts Modula-2 keywords to uppercase
Forth Breshehan circle algorithm example
Analyze 12 templates for the spreadsheet Analyze
There are four programs here that read Commodore 64 picture files. They can translate Koala Pad, Doodle, Print Shop and News Room graphics to IFF format. Getting the files from your C-64 to your Amiga is the hard part.

AMICUS Disk 12
Executable programs
blink 'alink' compatible linker, but faster, E-D
clean spins the disk for disk cleaners, E-D
epsonset sends Epson settings to PAR from menu E-D
showbig view hi-res pics in low-res superbitmap; E-D
speaktime tell the time, E-D
undelete undeletes a file, E-D
cnvapldhm converts Apple][low, medium and high res pictures to IFF, E-D
menued menu editor produces C code for menus, E-D
quick quick disk-to-disk nibble copier, E-D
quickEA copies Electronic Arts disks, removes protection, E-D
txed 1.3 demo of text editor from Microsmiths,E-D
C programs
spin3 rotating blocks graphics demo, S-E-D
popcli start a new CLI at the press of a button, like Sidekick, S-E-D
vsprite VSprite example code from Commodore, S-E-D
AmigaBBS Amiga Basic bulletin board prog., S-D
Assembler programs
star10 makes star fields like Star Trek intro,S-E-D
Pictures
Mount Mandelbrot 3D view of Mandelbrot set
Star Destroyer hi-res Star Wars starship
Rcbot robot arm grabbing a cylinder
Texts
vendors Amiga vendors, names, addresses
cardco fixes to early Cardco memory boards
cinclude cross-reference to C include files
mindwalker clues to playing the game well
slideshow make your own slideshows from the Kaleidoscope disk

Where can you find all the
Fred Fish Collection, as well as the
Amicus Disks and The AC Disks,
cross referenced and fully listed?

AC's GUIDE TO AMIGA

AC's Guide lists the descriptions and
contents of over 280
Freely Distributable Software
disks as well as

over 2200 Amiga products.

On Sale Now
at an your local Amazing Dealer

AMICUS Disk 13
Amiga Basic programs
Routines from Carolyn Scheppner of CBM Tech Support, to read and display IFF pictures from Amiga Basic. With documentation. Also included is a program to do screen prints in Amiga Basic, and the newest BMAP files, with a corrected ConvertFD program. With example pictures, and the SaveILBM screen capture program.

Routines to load and play FutureSound and IFF sound files from Amiga Basic, by John Foust for Applied Visions. With documentation and C and assembler source for writing your own libraries, and interfacing C to assembler in libraries. With example sound.

Executable programs
gravity — Sci Amer Jan 86 gravitation graphic simulation, S-E-D
Texts
MIDI — make your own MIDI instrument interface, documentation & a hi-res schematic.

AMICUS Disk 14
Several programs from Amazing Computing issues:
Tools
Dan Kary's — C structure index program, S-E-D
Amiga Basic programs:
BMAP Reader — by Tim Jones
IFFBrush2BOB — by Mike Swinger
AutoRequester — example
DOSHelper — Windowed help system for CLI commands, S-E-D
PETrans — translates PET ASCII files to ASCII files, S-E-D
C Squared — Graphics program from Scientific American, Sept 86, S-E-D
crlf — adds or removes carriage returns from files, S-E-D
dpdecode — decrypts Deluxe Paint, remo
ves copy protection, E-D
queryWB — asks Yes or No from the user returns exitt code, S-E
vc — VisiCalc type spreadsheet, no mouse control, E-D
view — views text files with window and
slider gadget, E-D
Oing, Sproing, yaBoing, Zoing are sprite-based Boing! style demos, S-E-D
CLIClock, sClock, wClock are window border clocks, S-E-D
Texts
An article on long-persistance phospor monitors, tips on making brushes of odd shapes in Deluxe Paint, and recommendations on icon interfaces from Commodore-Amiga.

AMICUS Disk 15
The C programs include:
'pr' — a file printing utility, which can print files in the background, and with line numbers and control character filtering.
'fm' — displays a chart of the blocks allocated on a disk.
'Ask' — questions an 'execute' file, returns an error code to control the execution in that batch file
'Stat' — an enhanced version of AmigaDOS 'status' command.
'Dissolve' — random-dot dissolve demo displays IFF picture slowly, dot by dot, in a random fashion.
'PopCLI2' — invoke new CLI window at the press of a key.
The executable programs include:
'Form' — file formatting program through the printer driver to select print styles
'DiskCat' — catalogs disks, maintains, sorts,merges lists of disk files
'PSound' — SunRize Industries' sampled sound editor & recorder
'Iconmaker' — makes icons for most programs
'Fractals' — draws great fractal seascapes and mountain scapes.
'3D Breakout' — 3D glasses, create breakout in a new dimension
'AmigaMonitor' — displays lists of open files,
memory use, tasks, devices and ports in use.
'Cosmoroids' — version of 'asteroids' for the Amiga.
'Sizzlers' — high resolution graphics demo written in Modula 2.

Texts:
'ansi.txt' — explains escape sequences the CON: device responds to.
'FKey' — includes template for making paper to sit in the tray at the top of the Amiga keyboard.
'Spawn' — programmer's document from Commodore Amiga, describs ways to use the Amiga's multitasking capabilities in your own programs.
AmigaBasic programs:
'Grids' — draw sound waveforms, and hear them played.
'Light' — a version of the Tron light-cycle video game.
'MigaSol' — a game of solitaire.
'Stats' — program to calculate batting averages
'Money' — *try to grab all the bags of money that you can.*
AMICUS 15 also includes two beautiful IFF pictures, of the enemy walkers from the ice planet in Star Wars, and a picture of a cheetah.

AMICUS Disk 16
'juggler' — demo by Eric Graham, a robot juggler bouncing three mirrored balls, with sound effects. Twenty-four frames of HAM animation are flipped quickly to produce this image. You control the speed of the juggling. The author's documentation hints that this program might someday be available as a product.
IFF pictures
parodies of the covers of Amiga World and Amazing Computing.
C programs:
'Inputhandler' — example of making an input handler.
'FileZap3' — binary file editing program
'ShowPrint' — displays IFF picture, and prints it.
'Gen' — program indexes and retrieves C structures and variables declared in the Amiga include file system.
Executable Programs:
'FixHunk2' — repairs an executable program file for expanded memory
'ms2smus' — converts Music Studio files to IFF standard 'SMUS' format. I have heard this program might have a few bugs, especially in regards to very long songs, but it works in most cases.
'Missile' — Amiga version of the 'Missile Command' video game,

This disk also contains several files of scenarios for Amiga Flight Simulator II. By putting one of these seven files on a blank disk, and inserting it in the drive after performing a special command in this game, a number of interesting locations are preset into the Flight Simulator program. For example, one scenario places your plane on Alcatraz, while another puts you in Central Park

AMICUS Disk 17
Telcommunications disk whuch contains six terminal programs.
"Comm" V1.33 — term prog. with Xmodem, WXmodem,
"ATerm" V7.2 — term prog. includes Super Kermit
"VT-100"V2.6 — Dave Wecker's VT-100 emulator with Xmodem,Kermit, and scripting
"Amiga Kermit" — V4D(060) port of the Unix C-Kermit
"VTek" V2.3.1 — Tektronix graphics terminal emulator based on the VT-100 prog. V2.3 and contains latest 'arc' file compression
"AmigaHost" — V0.9 for Compuserve. Includes RLE graphics abilities & CIS-B file transfer protocol.
"FixHunk" — expansion memory necessity
"FixObj" — removes garbage characters from modem received files
"Txt" — filters text files from other systems to be read by the Amiga E.C.
"addmem" — executeable version for use with mem expansion article in AC v2..1
'arc' — file documentation and a basic tutorial on un 'arc'ing files
"arcre" — for makeing "arc" files E.C.

AMICUS Disk 18
Logo — Amiga version of the popular computer language, with example programs, E-D
Tv*Text — Demo version of the TV*Text character generator

PageSetter — Freely distributable versions of the updated PagePrint and PageIFF programs for the PageSetter desktop publishing package.
FullWindow — Resizes any CLI window using only CLI commands, E-D
Life3d — 3-D version of Conway's LIFE program, E-D
Defdisk — CLI utility to re-assign a new Workbench disk, S-E-D
Calendar.WKS — Lotus-compatible worksheet that makes calendars
SetKey — Demo of keyboard key re-programmer, with IFF picture to make function key labels, E-D
VPG — Video pattern generator for aligning monitors, E-D
HP-10C — Hewlett-Packard-like calculator, E-D
SetPrefs — Change the Preferences settings on the fly, in C, S-E-D
StarProbe — Program studies stellar evolution. C source included for Amiga and MS-DOS, S-E-D
ROT — C version of Colin French's AmigaBasic ROT program from Amazing Computing. ROT edits and displays polygons to create three dimensional objects. Up to 24 frames of animation can be created and displayed. E-D
Scat — Like lng, windows on screen run away from the mouse, E-D
DK — Decays* the CLI window into dust, in Modula 2, S-E-D
DropShadow2 — Adds layered shadows to Workbench windows, E-D

AMICUS Disk 19
This disk carries several programs from Amazing Computing. The IFF pictures on this disk include the Amiga Wake part T-shirt logo, a sixteen-color hi-res image of Andy Griffith, and five Amiga Live! pictures from the Amazing Stories episode that featured the Amiga.
Solve — Linear equation solver in assembly language, S-E-D
Gadgets — Bryan Catley's AmigaBasic tutorial, S-D
Household — Bryan Catley's AmigaBasic household inventory program, S-D
Waveform — Jim Shields' Waveform Workshop: AmigaBasic, S-D
DiskLib — John Kennan's AmigaBasic disk librarian program, S-D
Subscripts — Ivan Smith's AmigaBasic subscript example, S-D
String, Boolean — C programs and executables for Harriet Maybeck Tolly's Intuition tutorials, S-E-D
Skinny C — Bob Riemersma's example for making small C programs, S-E-D
COMAL.h — Make C look like COMAL header file, S-D
EmacsKey — Makes Emacs function key definitions by Greg Douglas, S-D
AMon 1.1 — Snoop on system resource use, E-D
BTE — Bard's Tale character editor, E-D
Size — CLI program shows the size of a given set of files, E-D
WinSize — CLI window utility resizes current window, S-E-D

AMICUS Disk 20
Compactor, Decoder — Steve Michel AmigaBasic tools, S-D
BobEd — BOB and sprite editor written in C,S-E-D
SpriteMasterII — Sprite editor and animator by Brad Kiefer, E-D
BlitLab — Blitter chip exploration C program by Tomas Rokicki, S-E-D
FPic — Image processing program by Bob Bush loads and saves IFF images, changes them with several techniques, E-D
Banknk — Complete home banking prog., balance your checkbook! E-D

AMICUS Disk 21
Target — Makes each mouse click sound like a gunshot, S-E-D
Sand — Simple game of sand that follows the mouse pointer, E-D
PropGadget — Harriet Maybeck Tolly's proportional gadget example, S-E
EHB — Checks to see if you have extra-half-bright graphics, S-E-D
Piano — Simple piano sound program
CelScripts — Makes cel animation scripts for Aegis Animator, in AmigaBasic

This disk has electronic catalogs for AMICUS disks 1 to 20 and Fish disks 1 to 80. They are viewed with the DiskCat program, included here.

AMICUS Disk 22
Cycles — Light cycle game, E-D
Show_PrintII — Views and prints IFF pictures, including larger than screen
PrtDrvGen2.3 — Latest version of a printer driver generator
Animations — VideoScape animations of planes and boing ball
Garden — Makes fractal gardenscapes
BasicSorts — Examples of binary search and insertion sort in AmigaBasic

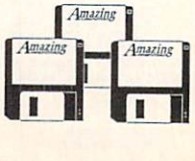
AMICUS Disk 23
An AMICUS disk completely dedicated to music on the Amiga. This disk contains two music players, songs, instruments, and players to bring the thrill of playing "Big Sound" on your Amiga
Instruments — a collection of 25 instruments for playing and creating music. The collection ranges from Cannon to Marimba
List INSTR — program to list the instruments DMCS will not load as well as list the origins for any instrument.
Music 1812Overture — a collection of 14 Classical pieces — The 16 minute classical feature complete with Cannon!
Three Amiga Music Players: SMUSPlay, MusicCraft2SMUS, MusicStudio2SMUS

AMICUS Disk 24
Sectorama — A disk sector editor for any AmigaDOS file-structured device, recover files from a trashed hard disk. By David Joiner of Micrillusions
Iconize — Reduces the size of IFF images, companion program, Recolor, remaps the palette colors of one picture to use the palette colors of another. Using these programs and a tool to convert IFF brushes to Workbench icons, make icons look like miniatures of the pictures.
CodeDemo — Modula-2 program converts assembler object files to inline CODE statements. Comes with a screen scrolling example
AmiBug — Workbench hack makes the same fly walk across the screen at random intervals. Otherwise, completely harmless.
BNTools — Three examples of assembly language code from Bryce Nesbitt:
1. SetLace.prog to switch interlace on&off.
2. Why, replace AmigaDOS CLI Why
3. LoadIt, prog to load a file into memory until a reboot. (Only the most esoteric hackers will find LoadIt useful.)
Monolace — CLI program resets Preferences to several colors of monochrome & interlace screens. C source is included, works with DisplayPref, a CLI program which displays the current Preferences settings.
BoingMachine — A ray-traced animation of a perpetual motion Boing-making machine, includes the latest version of the Movie program, which has the ability to play sounds along with the animation. By Ken Offer
Daisy — Example of using the translator and narrator devices to make the Amiga talk. It is written in C.
QuickFlix — Script-driven animation and slideshow program flips through IFF images.

BMon — System monitor AmigaBasic program ; perform simple manipulations of memory.
Moose — Random background program, a small window opens with a moose resembleing Bullwinklesaying witty phrases user definable.
DGCS — Deluxe Grocery Construction Set, simple Intuition-based prog for assembling and printing a grocery list.
The Virus Check directory holds several programs relating to the software virus that came to the US from pirates in Europe as detailed in Amazing Computing V2.12. Bill Koester's full explanation of the virus code is included. One program checks for the software virus on a Workbench disk; the second program checks for the virus in memory, which could infect other disks.

AMICUS Disk 25
Nemesis — Graphics demo pans through space towards the mythical dark twin of the sun with wonderful music and space graphics.
The KickPlay directory holds text that describes several patches to the Kickstart disk. For Amiga 1000 hackers who feel comfortable patching a disk in hexadecimal, KickPlay offers the chance to automatically do an ADDMEM for old expansion memory, as well as the ability to change the picture of the "insert Workbench" hand. A program is also included for restoring the correct checksum of the Kickstart disk.
KeyBird — BASIC prog edits keymaps, adjust the Workbench keymaps or create your own.
8ColorWB — Modifies the Workbench so three bitplanes are used, icons can have eight colors, instead of four. eight-color icons are included. Public domain program "zapicon" or "brush2icon" converts eight-color IFF brushes to icons, to use Deluxe Paint to make icons for this new Workbench.
BrushIcon — Converts brushes to icons (bizarr docs).
Egraph — Graphing prog reads [x,y] values from a file and displays them on the screen, similar to the same-named Unix program.
Keep 1.1 — Message-managing program for telecommunications, lets you save messages from an online transcript to another file, understands the message format of the national networks and several types of bulletin board software. Moves through the transcript and save messages
Kill.fastdir — Speed up directory access, it creates a small file in each directory on a disk which contains the infor-mation about the files, will also remove all the "fastdir" files from each directory. by CLImate's authors
The LaceWB program changes between interlace and non-interlace Workbench. Previously, you were forced to reboot after changing Preferences to an interlaced screen. This program flips between the normal and extended screen heights.
PW_Utility — A shareware utility for ProWrite users, changes margin settings and font types.
Guru — A CLI program, prints out probable causes for Guru mediations; C source included.
DiskWipe — Latest from Software Distillery, removes files from directories or disk drives, much faster than "delete."
Snow — AmigaBasic makes snowflake designs.
Mlist — Mailing list database.
Softballstats — Maintain softball statistics/ team records.
Dodge — Short Modula-2 program moves the Workbench screen around after a period of time, prevents monitor burn-in.

AMICUS Disk 26
Todor Fay's SoundScape module code from his Amazing Computing articles. The source to Echo, Chord, TX, and VU is included. The Lattice and Manx C source code is here, along with the executable modules.
Claz2 — Update of prog to convert IFF images to PostScript files for printing on laser printers
SDBackup — Hard disk backup prog with Lempel-Ziv compression to reduce the necessary number of disks.
TCB — Prints information about tasks and processes in the system; assembler source is included.
FunBut — Lets a function key act like a rapid series of left mouse button events.
DC — A handy program for people who use an Amiga 1020 5 1/4 inch drive as an AmigaDOS floppy. A Workbench program that sends a DiskChange signal to the operating system: instead of typing "diskchange df2:" over and over again, just click on the icon. C source included.
System config — File makes screen 80 columns wide of text in the Scribble! word processor.
Dick2Ram — 2 programs to move the Scribble! spelling dictionary to and from the RAM disk.
Lexical — Analyzes a text file and gives the Gunning-Fog, Flesch, and Kincaid indices which measure readability.
HexDump — Modula-2 program to display memory locations in hexadecimal.
Tartan — AmigaBasic; design Tartan plaids.
DirMaster — Disk catalog program.
BMP — plays 8SVX sampled sounds in the background while something else is happening in the Amiga, as your Amiga is booting, for example.
ShowPt — CLI program changes your pointer to a given pointer.
AMICUS 26 also has a collection of mouse pointers, & Workbench program to display them



For PDS orders, please use form on page CIII
Visa and MasterCard is available on orders of $20.00 or more.

The Fred Fish Public Domain Software Library

The Fred Fish disks are collected by Mr. Fred Fish, a good and active friend of the Amiga.

Fred Fish Disk 232

BallyIII Amiga port of the former arcade game named Click. This version fixes some minor bugs and is faster than the previous versions. This is version III, an update to the version released on disk 221. Binary only, shareware. Author: Oliver Wagner

Dbug Machine independent macro based C debugging package. Provides function trace, selective printing of internal state information, and more. This is an update to the version released on disk 102, and now includes a machine independent stack use accounting mechanism. Includes source. Author: Fred Fish; profiling support by Binayak Banerjee

ReSourceDemo A demo version of ReSource, an interactive disassembler for the Amiga. This is a complete version except that the "save" features have been disabled. This is version 3.06, an update to version 0.36 from disk 192. Binary only. Author: Glen McDiarmid

Fred Fish Disk 233

Brik A general purpose program that calculates both text and binary cyclic redundancy codes (CRCs). Text mode CRCs calculated by brik are portable across systems for files that are in the usual text format on each system. Binary mode CRCs are portable for files that are moved from system to system without any change. Brik can be used to verify and update an embedded checksum header in files. It runs under MS-DOS, UNIX system V, BSD UNIX, VAX/VMS, and AmigaDOS. This is version 2.0 and includes source. Author: Rahul Dhesi

CacheCard An accessory to SetCPU for use with A2620 cards or 68030 systems. It modifies the MMU table set up by SetCPU to selectively control caching for each expansion card. It's also an example of how an accessory program can track down and modify the SetCPU MMU table without having to read all kinds of MMU registers and figure it out for yourself. Version 1.00, includes source. Author: Dave Haynie

CrcLists Complete CRC check files for disks 001-231 using the brik program also on this disk. These were made directly from my master disks. I have switched to brik, from the crc program used to make the lists on disks 133, 146, and 173, because it has more features and because source is available. Author: Fred Fish

Fred Fish Disk 234

KwikBackUp A harddisk backup program that writes data track by track onto multiple floppy disks. Uses the archive bit, saves and restores comments and protection flags, and skips over bad spots during restore. Version 1.0, includes source in Modula-II. Author: Fridtjof Siebert

MuchMore Another program like "more", "less", "pg", etc. This one uses its own screen to show the text using a slow scroll. Includes built-in help, commands to search for text, and commands to print the text. Works with PAL or NTSC, in normal or overscan modes. Supports 4 color text in bold, italic, underlined, or inverse fonts. Version 1.8, includes source in Modula-II and assembly code. Author: Fridtjof Siebert

NetWork Another program in the long tradition of "screen hacks" for the Amiga. Won't spoil the surprise by saying what it does. Version 1.0, includes source in Modula-II. Author: Fridtjof Siebert

Printit A program to print IFF pictures on Epson compatible 9-pin printers. Prints in many resolutions, with many ways to convert color pics to black and white. Version 1.0, includes source in Modula-II. Author: Fridtjof Siebert

WBPic Replaces Workbench's color 0 with an IFF hires non-interlaced picture, in 2 or 4 colors. Version 1.0, includes source in Modula-II. Author: Fridtjof Siebert

XHair Replaces the mouse pointer with a screen wide crosshair, which is useful for positioning things vertically or horizontally. Version 1.0, includes source in Modula-II. Author: Fridtjof Siebert

Fred Fish Disk 235

CalcKey A basic four function, memory resident, pop-up calculator which uses only about 24K of memory and can automatically type the answer to any calculation into the program you were using when it was popped up. Version 1.0, binary only, shareware. Author: Craig Fisher

Ct An Amiga program to display images from a CT scanner, along with several new interesting sample images of scans of real people. The display software, though it has a primitive user interface, is quite powerful, including functions like convolutions, averaging, laplacians, unsharp masking, edge detection, gradients, etc. This is version 2.2, an update to the version on disk 137. Binary only. Additional image disks available from author. Author: Jonathan Harman

MirrorWars A new game featuring sound, title music, and two player mode. You fight your opponent via laser rays, but beware of the mirrors reflecting your shots. Binary only. Author: Oliver Wagner

Fred Fish Disk 236

AmigaBench Optimized Amiga assembly versions of the Dhrystone benchmark. Includes 68000 and 68020 versions. Author: Al Aburto

DiskHandler A sample implementation of a file system that reads and writes 1.2 format diskettes. Includes source. Author: Software Distillery

Heart3D A program to find left ventricle outlines in the output of an Imatron CT scanner, and display wireframe animations of the beating heart. Includes several sample CT scan outputs. Binary only. Author: Jonathan Harman

Ls Version 3.1 of the popular UNIX style directory lister. This is an update to version 2.0 from disk 178, and includes some bug fixes, support for multiple wildcard pathnames, quicker sorting, a best-fit output, new output width and height options, and some other new features. Includes source. Author: Justin V. McCormick.

Proc Example program of how to create a full-fledged DOS process without needing to call LoadSeg first. Based on an idea presented at BADGE. Includes source. Author: Leo Schwab

XprZmodem An Amiga shared library which provides ZModem file transfer capability to any XPR-compatible communications program. Version 1.0, includes source. Author: Rick Huebner

Fred Fish Disk 237

CLIPrint An example of printing to the CLI from assembly code. Includes source (of course). by: Jeff Glatt

CType Another text file reader, but this one is small, reasonably fast, and includes bi-directional scrolling, search, go to a given percentage, and printing capabilities. Version 1.0, includes source in assembly. Author: Bill Nelson.

StripCR This little program just makes a text file ready for use with AmigaDOS, with only LineFeed characters (LF) to mark the end of a line. If you feed it a file with ONLY Carriage Return characters (CR), (from a Macintosh for example) it will replace them with the LF character and, if the file requires no changes, then it does not get changed, includes source in assembly. by: Bill Nelson

PlusCR Companion program to StripCR, it reverses the procedure. PlusCR produces a file ready for use on systems which require both the CR and LF characters to mark the end of a line (such as those running MS-DOS for example, includes source in assembly. Author: Bill Nelson

StripLF Completes the set of StripCR and PlusCR. It will change an LF only file into a CR only file. If used in combination with StripCR and PlusCR, it completes a text file conversion system, includes source in assembly. Author: Bill Nelson

CLS Clear Screen Command made for the purposes of being SMALL, and thus not wasteful of memory when made memory resident, it consists of 96 bytes of memory on disk! Includes source in assembly. Author: Bill Nelson

DPlot A simple display program for experimental data, with the goals of supporting paging through lots of data and providing comfortable scaling and presentation. Version 1.0, source available from author. Author: A. A. Walma

ILBMLib A shared library (ilbm.library) to read/write IFF files, derived from the EA IFF code, along with various enhancements. Includes examples of using the library from C code, assembly code, or BASIC, along with source for examples and interface code. Author: Jeff Glatt

ParOut Shows how to allocate and communicate directly with the parallel port hardware from an assembly language program. Includes source. Author: Jeff Glatt (original C code by Phillip Lindsay)

Speed A performance benchmark useful for comparing Amiga processing speeds. Performs 10000 iterations of some selected groups of 68000 instructions while using the DateStamp time function to record how many ticks it takes to complete. This timed duration is then compared against two known prestored times, one for a stock A2000 Amiga and one for an A2620 enhanced A2000. A relative comparison is calculated and displayed. Version 1.0, includes source in assembly language. Author: Jez San

Fred Fish Disk 238

CWDemo Demo version of a pop-up utility to control the color register assignments of Intuition custom screens. Version 3.1, binary only. Author: Kimbersoft

DMouse A versatile screen & mouse blanker, auto window activator, mouse accelerator, popcli, pop window to front, push window to back, etc, widget. Includes DLineArt, a screen blanker replacement program for use with DMouse. This is DMouse version 1.20, an update to version 1.10 on disk 168/169. Includes source. Author: Matt Dillon

LabelPrint A program that allows you to easily print labels for your disks. This is version 2.5, an update to version 1.9 from disk 210. Shareware, binary only (source available from author). By: Andreas Krebs

NGC Yet another virus check program. Checks the bootblock on all inserted floppy disks and reports nonstandard ones. Checks the jump tables of all resident libraries and devices and reports suspicious entries. Version 1, includes source in assembly. Author: Ulf Nordquist

Pyth A program to draw the Tree of Pythagoras. Version 1.1, includes source. By: Andreas Krebs

Steinschlag A tetris like game (Steinschlag means "Falling Rock") submitted by the author. This is version 1.8, an update to version 1.5 from disk 221. Binary only. Author: Peter Handel

Fred Fish Disk 239

FF 239 contains Forth programs from the JGoodies #1 disk, from Delta Research (the makers of JForth Professional 2.0). All of the material has been placed into a subdirectory (JGoodies). Below is a listing of subdirectories under JGoodies, and their contents.

Brunjes Various tools submitted together by the author. StringPkg is string package for both Forth style and NUL terminated strings. Date&Time are handy tools for getting and printing formatted date and time. Utils are utilities used by the other files. CursorControl is an example of moving the text cursor. SpaceOrEscape is a handy word for pausing or stopping program output. Includes source code. Author: Roy Brunjes

Evolution This program graphically simulates the evolution of a species of "bugs", the insect kind. Bugs, represented by moving blobs, eat bacteria represented by single pixels. They mutate, compete for food, reproduce and pass their mutations to their offspring. Fascinating example of graphics and software simulation. Standalone image and source code. Author: Russel Yost

FFT Highly optimized Fast Fourier Transform tools for digital signal processing. The FFT can be used to compute the frequency spectrum of a complex signal. It is useful in a variety of different applications. Floating point and integer versions. Mixture of high level and assembly language code. Includes source (requires JForth). Author: Jerry Kallaus

Guru Handy "guru" number interpreter (well, handy after reboot anyway!). Tells you what "81000009" means, for example. CLI usage only. Standalone image with readme file. Source code included. Author: Mike Haas

H2J Converts 'C' style '.h' include files to JForth style '.j' files. Useful when developing interfaces to new Amiga libraries like ARP, etc. Standalone image and source code. Author: Phil Burk

HAMmmm2 Graphics hack that displays moving lines in a HAM screen for a hypnotic effect. Uses sound tools from HMSL, if available, for a drone sound that corresponds to the graphics image. Standalone image and source code. Author: Phil Burk

HeadClean This program, combined with a fibre cleaning disk, can be used to clean the heads on your disk drives. Source code examples of accessing the Trackdisk device, and using gadgets are included. Standalone image with source code. Shareware. Version 2.0. Author: Phil Burk

JustBeeps Simpie example of using Audio and Timer devices. Plays a series of beeps whose pitches are based on a just intoned tuning system. Standalone image with source code. By: Phil Burk

Mandelbrot A fast Mandelbrot rendering program that uses some of the mathematical properties of the Mandelbrot set to greatly reduce the drawing time. Demonstrates graphics programming, assembly language, menus and IFF file I/O. Standalone image with source code. Author: Nick Didkovsky

NeuralNet Example of Neural Net programming converted to JForth. Demonstrates a programming technique that many say is the wave of the future for software. This is a simple demo that shows neural propagation. Standalone image with source code. Author: Robert E. La Quey, ported by Jack Woehr

Textra This easy-to-use text editor allows multiple windows, and provides a simple mouse driven interface. Those familiar with the "Macintosh style" editors will be comfortable with Textra's Cut, Copy and Paste commands. Standalone image. Documentation included. No source code. Author: Mike Haa

Fred Fish Disk 240

CrossDOS A "tryware" version of a mountable MS-DOS file system for the Amiga. This is a software product that allows you to read and write MS-DOS/PC-DOS and Atari ST formatted disks (Version 2.0 or higher) directly from AmigaDOS. This tryware version is a "read only" version, which does not allow any writes to the disk. A fully functional version is available for a very reasonable price from CONSULTRON. Version 3.02, binary only. Author: CONSULTRON, Leonard Porna

Dis An AmigaDOS shareable library which implements a symbolic single-instruction disassembler for the MC68000 family and a program which uses the library to disassemble/dump AmigaDOS object files, making full use of symbolic and relocation information. Includes source code in Draco. Author: Chris Gray

DM-Maps IFF maps to the Dungeon Master game. All 14 levels are included. Author: Unknown

MemLib A link library of routines to aid in debugging memory problems. Works with Lattice C 5.0 and possibly with earlier versions. It's features include trashing all allocated memory, trashing all freed memory, keeping track of freed memory with notification if it is written to, notification of memory freed twice or not at all, notification of overrunning or underrunning allocated memory, generation of low memory conditions for testing purposes, and identification of violations of memory use by filename and line number of the allocating routine. Includes source. Author: John Toebes and Doug Walker

RunBack Allows you to start a new CLI program and run it in the background, then closes the new CLI. This is version 6, an update to the version on disk 152 (the version on disk 214 appears to be on a different evolutionary path). This version compiles under Lattice with many optimizations enabled, and can be made resident. Includes source. Author: Rob Peck, Daniel Barrett, Greg Searle, Doug Keller

XprLib External file transfer protocol library. Document and code example for implementing external file transfer protocols using Amiga shared libraries. This is an update to the version included with the vit program on disk 226. By: Willy Langeveld

Fred Fish Disk 241

ASDG-rrd Extremely useful shareware recoverable ram disk. This AmigaDOS device driver implements a completely DOS compatible disk device in memory that survives resets, guru's, and crashes. An absolute must for those with lots of ram. This is an update to the version released on disk 58. It now works with up to 8Mb of memory. It was rewritten in assembly and is now faster and much smaller. Binary only. Author: Perry Kivolowitz, ASDG Inc.

CBBS The W0RLI BBS system for use in amateur radio. Originally written for IBM-PC compatibles, it was ported to the Amiga by Pete Hardie. This is version 6.1c with source code. Author: Hank Oredson, the CBBS group, Pete Hardie

Fix68010 A program which patches executables that fail to run on machines equipped with an M68010, so that they no longer use the prohibited priviledged instructions. Binary only. Author: Gregor Brandt

Man A program similar to the UNIX "man" program. Displays information about a topic from manual pages. Does not include any database of topics, you have to supply your own. Version 1.2, includes source. Author: Garry Glendown

NoClick A program which silences the clicking of empty drives on the B2000 under AmigaDOS 1.3. It should also work on an A500. This is version 3.4, an update to the version on disk 231. Includes assembly source code. Author: Norman Iscove

Tiles A basic tile game like Shanghai or GunShy. A board is covered with a set of 144 tiles, 36 different sets of 4 identical tiles, each with a picture on it. The object is to remove all the tiles, 2 at a time, by matching identical tiles. Version 2.1, includes source in Modula II. Author: Todd Lewis

Fred Fish Disk 242

BootBlocks Detailed documentation on what a bootblock is and how it works, along with some sample bootblocks and a program to install a custom bootblock. Includes source for the sample bootblocks and the install program. Author: Jonathan Potter

Check4Mem Allows you to check for a specified amount of memory, with certain attributes, from a batch file. If the requirements are not met, a WARN returncode is generated. Version 2, includes source. Author: Jonathan Potter

CustReq A glorified ASK command for your startup-sequence. It generates a requester with the specified title, text, positive and negative gadgets (either of which can be the default), and an optional timeout value. Version 2, includes source. Author: Jonathan Potter

FileReq This is Jonathan's second version of a file requester, and is much more powerful than the one included on disk 204. Shareware, includes source. Author: Jonathan Potter

FullView A text viewer that uses gadgets at the bottom of the screen (thus can display text 80 columns wide), opens up to the full height of the workbench screen, has fast scrolling, and can work with compressed files (file compression program included). Shareware, binary only, source availablefrom author. By: Jonathan Potter

Image-Ed An icon editor that allows you to draw and edit images up to 150 by 90, in up to 16 colors. Allows freehand drawing, empty or filled rectangles, ellipses, and triangles, lines curves, and polygons, copy, flip about x or y axis, stretching and condensing, flood fill and complement, text with selection and loading of font style, undo, magnified and normal sized images, and two active drawing screens at once. This is version 2.2, an update to version 1.9 on disk 211. Binary only, source available from author. Author: Jonathan Potter

JAR A shareware game (Jump And Run) using 3-D graphics. Your task is to collect the blue pills lying on the floors and steps, not to fall down or off the steps, and to avoid several monsters wandering about. You can collect various sorts of weapons to use against the monsters. Version 1.0, binary only, source available from author. Author: Andreas Ehrentraut

JPClock A short clock program that is just packed with features. This is version 1.2, an update to version on disk 204. Includes source. by: Jonathan Potter

PPrefs Preferable Preferences is a program designed to replace the standard preferences, that is shorter, more efficient, and easier to use. Binary only. Author: Jonathan Potter

PaletteReq An easy way to set the palette of any screen from your program. Includes source. by: JonathanPotter

PopInfo A small utility which "pops open" to give you information about the status of your devices and memory. This is version 3.1, an update to version 3.0 on disk 223. Includes source. Author: Jonathan Potter

ZeroVirus A fully integrated virus checker and killer, with bootblock save and restore features. Finds both bootblock and file based viruses. This is version 1.3, binary only. Author: Jonathan Potter

Fred Fish Disk 243

FragIt A dynamic memory thrasher for the Amiga. FragIt randomly allocates and deallocates psuedo-random size values of memory, ranging from 16 bytes to 50000 bytes by default. The result is an allocation nightmare, thousands of memory fragments are being created and destroyed continuously. This puts stress on the memory allocation routines of an application undergoing testing by simulating a very busy, highly fragmented memory environment. This is version 2.0, featuring many bug fixes, a full intuition interface, configuration settings via the icon, and more. Includes source. By: Justin V. McCormick

ImageLab A program which performs image processing on IFF pictures. Includes standard image processing functions such as convolution, averaging, smoothing, enhancement, histograms, FFT's, etc. Also includes file conversion functions, a clipboard, and other useful functions. Version 2.2, binary only. Author: Gary Milliorn

LPE LaTeX Picture Editor is a graphical editor for producing "pictures" for the LaTeX system, which may be imported by LaTeX. You can draw boxes, dashed boxes, lines, vectors, circles, boxes with centered text, and plain text. This is version 1.0, binary only. Author: Joerg Geissler

NoClick A program which silences the clicking of empty drives on the B2000 under AmigaDOS 1.3. It should also work on an A500. This is version 3.5, a last minute update to FF241. Includes assembly source code. By: Norman Iscove

Password A program which enhances your computer's security by making it complicated enough that users without your password will get discouraged trying to boot and use your system. This should keep out most casual or nontechnical users. Version 1.21p, binary only. Author: George Kerber

Pcopy An intuition based disk copier similiar to the resident "DiskCopy". This is version 2.0, a highly upgraded rewrite of the version on disk 151. It features high speed diskcopy with write-verify and data recovery from damaged tracks. A lot of effort has gone into making this copier friendly in its usage, as well in its multitasking properties. Binary only. Author: Dirk Reisig

SimGen This program will add a 2 or 4 color picture to your WorkBench screen. If the picture is digitized, it will look much like a genlock, hence the name SimGen (Simulated Genlock). Binary only. By: Gregg Tavares

SuperLines A new lines demo with a realtime control panel that you can use to change various aspects of the action. Has 10 built in color palettes, support for things like color "smudge", color cycling, color "bounce", multiple resolutions, and can display either lines or boxes. This is version 1.0, binary only. Author: Chris Bailey

WarpUtil Warp (version 1.11), UnWarp (version 1.0), and WarpSplit (version 1.1). Warp reads raw filesystems and archives them into a compressed version in a normal file. UnWarp turns them back into filesystems. WarpSplit splits them up into smaller pieces on a track by track basis. Binary only. Author: SDS Software

Fred Fish Disk 244

BBChampion This is BootBlockChampionIII, a very nicely done program that allows you to load, save, and analyze any bootblock. This is version 3.1, binary only. Author: Roger Fischlin

BootIntro This program creates a small intro on the bootblock of any disk, which will appear after you insert the disk for booting. The headline can be up to 44 characters. The scrolling text portion can be up to 300 characters. This is version 1.2, an update to version 1.0 on disk 188. Binary only. Author: Roger Fischlin

FMC An alternative to the NoFastMem program. Uses a cute little switch gadget to turn fast memory on or off. Version 1.2, includes source in assembly code. Author: Roger Fischlin

SizeChecker Size checker uses a list of possible sizes of a file to check for unexpected changes in the size of those files. For example, it can be used to spot a link virus or to point out changes in the configuration of your system. With the appropriate comments added to your size list, you can check to see what version of the files you are using (1.2, 1.3, 1.4, ARP, etc). Version 1.0, binary only. Author: Roger Fischlin

TextDisplay A text display program, like "more" or "less", but about half the size and handles all screen formats (pal/ntsc, interlace/non-interlace, etc). This is version 1.52, an update to version 1.1 on disk 188. Binary only. Author: Roger Fischlin

XColor A program designed to change the colors of any screen. You can also add and subtract bitplanes in the screen, or convert the screen to black and white (grayscale). Handles HAM and EHB screens. Version 1.2, includes source in assembly code. Author: Roger Fischlin

Fred Fish Disk 245

ATOF A small utility that allows you to use the fonts of another disk without using the CLI. Version 1.0, binary only. Author: Roger Fischlin

BootIntro This program creates a small intro on the bootblock of any disk, which will appear after you insert the disk for booting. This is the "next generation" BootIntro, a more colorful version than the one on disk 244, but the text must be shorter. The first line can be up to 24 characters. The second and third lines can be up to 22 characters. The scrolling text can be up to 98 characters. This is "next generation version 1.1". Binary only. Author: Roger Fischlin

Fenster A program which can operate on windows owned by another program, to close them, change their size, refresh gadgets, move the window to the background, etc. Version 1.0, includes source. Author: Roger Fischlin

PathMaster A file requester with lots of features. Can be easily configured by the programmer to suit a variety of applications. Includes source. Author: Justin McCormick

Reversi Plays the classical reversi game on an 8 x 8 square field. Version 1.2, includes source in assembly code. Author: Marc Fischlin

Vlt This is a binary update to the vlt program on disk 226, and fixes a problem with external protocol support. You still need the rest of the files from disk 226. Version 4.065, binary only. Author: Willy Langeveld

Fred Fish Disk 246

Dmake Release version 1.0 of Matt's version of the UNIX make utility. Update to beta version released on disk 179. Features multiple dependancies, wildcard support, and more. Binary only. Author: Matt Dillon

LabelPrint A program that allows you to easily print labels for your disks. This is version 2.5b, an update to version 2.5 from disk 238. Shareware, binary only (source available from author). Author: Andreas Krebs

NComm A terminal program for the Amiga based on comm version 1.34. Has hot keys for most program functions (including dialing up to 10 phone numbers), PAL and NTSC support for normal or interlaced screens, screen I/O greater than 2400 baud, ANSI/VT100 terminal emulation with full 8 color text support, IBM graphics, optional translation styles, split screen mode, full user control of color palette, full support for all European languages, full serial port control with baud rates up to 19200, script language, phonebook, keyboard macros, and more. Version 1.8, binary only. Author: DJ James, Daniel Bloch, et. al.

NoClick A program which silences the clicking of empty drives on the B2000 under AmigaDOS 1.3. It should also work on an A500. This is version 3.6, an update to version to version 3.5 on disk 243. Includes assembly source code. Author: Norman Iscove

ScreenShare A library and support programs that enable applications to open up windows on other applications' custom screens. For example, your editor may want to open a window on your terminal emulator's screen so you can compose a message while still being able to see the contents of the terminal's screen. Both applications must cooperate for the screen sharing to work. Version 1.2, includes source for interface portions. Author: Willy Langeveld

Ty A text display program based on Amiga "less" version 1.0. Has both keyboard and mouse control of all functions, an intuition interface, and uses the Amiga specific keys (such as the Help key) correctly. Version 1.3, includes source. Author: Mark Nudelman, Bob Leivian, Tony Wills

Fred Fish Disk 247

AnalogJoystick Software support for use of analog joysticks on the Amiga. Includes a driver, a header file for code that calls the driver, and an example program that uses the driver. Includes source. Author: Dave Kinzer

AssemTools A collection of files which should be of great interest to Amiga assembly language programmers. The collection includes 140 macro routines which make assembly language programming a lot easier. There is also a library of routines containing buffered C-like file handling functions (fopen, fclose, fgets, fputs, etc) and a file name requester. All routines are re-entrant. Includes source for example programs using the macros and library, and a 65c02 cross macro assembler. Author: Jukka Marin

RemoteLogin A couple of programs which make possible remote access to your Amiga. One program checks the serial port for an incoming call, and starts a program when it is detected. The other is a password protection program which allows starting various programs based upon login id, thus providing some minimal security for your computer. Includes source. Author: Dave Kinzer

XprLib External file transfer protocol library. Document and code example for implementing external file transfer protocols using Amiga shared libraries. This is version 2.0, an update to the version on disk 240, with many extensions and enhancements. Includes sample XPR library and source. Author: Willy Langeveld

Fred Fish Disk 248

AmiGantt A project definition and management tool designed to create a simple, interactive method of outlining the task required to complete a particular project, using the GANTT chart as the input format. AmiGantt displays the project in a multi-window mode with separate windows for the GANTT chart, task information input, resource histogram display, and Pert chart display. Up to 500 tasks may be defined for any project, and a project may contain other projects as tasks. Version 3.0.0, shareware, binary only. Author: Donald Tolson

CLI-Colors A simple little program to change the colors of the border around a CLI window. Includes source in assembly code. Author: Michael Sinz

Flipper Small, fast, Othello program. Does not use any look-ahead methods. Binary only. By Michael Sinz

Maze An example of a fully re-entrant maze generation program written in assembly language using Manx's assembler. Includes source. Author: Michael Sinz

NetHandler The Software Distillery's network file system handler (NET:), using Matt Dillon's DNET to mount one Amiga's devices on another Amiga. It also serves as an example file system written entirely in Lattice C. Version 1.0, includes source. Author: Software Distillery

Regex An Amiga shared library version of the GNU regular expression package from the Free Software Foundation. A regular expression is a concise method of describing a pattern of characters in a string. By use of special wildcards, almost any pattern can be described. A regular expression pattern can be used for searching strings in such programs as editors or other string handling programs. Version 1.0, includes source. Author: Edwin Hoogerbeets, FSF, Jim Mackraz

Fred Fish Disk 249

Automata Four cellular automata programs. AutomaTron is a one dimensional cellular automaton, Crud is a automation based on a sum index rule, Demon is a cyclic space automation described in the Aug 89 Scientific American, and Life is one of the oldest and best known of all cellular automata. Includes source. Author: Gary Teachout

Slicer Slicer computes and displays images of the Mandelbrot and Julia sets. Unlike many Mandelbrot programs that generate pictures directly, Slicer computes and stores an array of raw data which it may then render into pictures in a number of ways. Version 1.0, binary only. Author: Gary Teachout

TurMite A two dimensional turing machine simulator. Imagine a small bug crawling around on your computer display moving one pixel at a time. At each step it uses its internal state number and the color of the pixel it's on as indexes into a set of tables to decide what color to change the pixel to, what direction to move, and what its new internal state should be. Source. By: Gary Teachout

Fred Fish Disk 250

ASimplex An implementation of the Simplex algorithm for solving linear programs. It uses the standardized MPSX-format for input data files. This is version 1.5, an update to version 1.2 on disk 199. Changes include bug fixes, the ability to run from CLI, it's own window for I/O, and some new and improved commands. Includes source. Author: Stefan Forster

Gravity-Well A celestial motion simulator that simulates the motion of up to twenty bodies in a Newtonian universe. The view of the simulation may be scaled, rotated in three dimensions or repositioned. Includes source. Author: Gary Teachout

Paranoids An asylum escape game. Paranoids is a traditional board game played by drawing cards, rolling dice, and moving pieces around the board. Each player has six pieces, four patients and two doctors. The object of the game is to get all of your patients out of the asylum. This is version 1.0, binary only. Author: Richard Anderson and Gary Teachout

RPSC A reverse polish scientific calculator. RPSC is a programmable RPN calculator in the Hewlett-Packard tradition. It supports operations with real numbers, complex numbers, matrices, and 3-D vectors, as well as storage and recall of labeled variables. Data and programs may be saved, loaded, or written as ASCII text, to AmigaDOS files. V1.1, binary only. By: Gary Teachout

Fred Fish Disk 251

Debug A symbolic debugger for the Amiga. Features include symbolic disassembly, simple expression parsing, multiple breakpoints, memory fill, memory dump, single step in trace mode, skip over subroutines, etc. V2.10, binary only. Author: Jim Thibodeau and Larry LaPlume

DiskSalv A disk recovery program for all Amiga file system devices that use either the Amiga Standard File System or the Amiga Fast File System. Disksalv creates a new filesystem structure on another device, with as much data salvaged from the original device as possible. This is version 1.42, an update to the version released on disk 212. Binary only. Author: Dave Haynie

DiskSpeed A disk speed testing program specifically designed to give the most accurate results of the true disk performance of the disk under test. Automatically updates and maintains an ASCII database of disk results for tested disks. This is version 1.0, includes source in C. Author: Michael Sinz

Linstrom Some miscellaneous small programs from the Linstrom family. Includes some basic games, some programs based on algorithms from the Computer Recreations column in Scientific American (demon, turmites, mountain), and a fish tank simulator. Includes source. Author: Lloyd Linstrom and family

Monopoly A simulation of the Parker Brother's board game MONOPOLY. Binary only. source available from author. Author: Ed Musgrove

RoadRoute A trip planner that takes a list of cities and a list of known routes between cities, and generates the distance and time required to reach your destination. V1.0, Source. Author: Jim Butterfield

Fred Fish Disk 252

CrossDOS A "tryware" version of a mountable MS-DOS file system for the Amiga. This is a software product that allows you to read and write MS-DOS/PC-DOS and Atari ST formatted disks (Version 2.0 or higher) directly from AmigaDOS. This tryware version is a "readonly" version, which does not allow any writes to the disk. A fully functional version is available for a very reasonable price from CONSULTRON. This is version 3.05b, an update to version 3.02 on disk 240. Binary only. Author: CONSULTRON, Leonard Poma

TrekTrivia Very nice mouse-driven trivia type program for Star Trek fans. Contains 100 questions with additional trivia disks available from the author. This is version 2.0, an update to the version on disk 180, and includes selectable skill levels, a cheat mode, and 250K of digitized music. Binary only, shareware. Author: George Broussard

Zerg An Ultima clone that takes your character through a simple but deadly fantasy world where you get to slaughter monsters, talk with the inhabitants of fully populated towns, embark on royal quests, purchase a plethora of weapons, cast spells, and otherwise occupy yourself while your computer is engaged in more important business. Version 1.0, binary only. Author: Mike Shapiro

Fred Fish Disk 253

DumpDiff A small utility that prints the hexdumps of 2 files, with all differences between these two files highlighted. Version 1.0, includes source in Modula-II. Author: Fridtjof Siebert

Elements Very nice interactive display of the the Periodic Table of Elements. Can display a large amount of pertinent data about a selected element along with a good deal of general and miscellaneous info. This is version 1.2a, an update to the version on disk 175. It loads faster, takes up less disk space, fixes a minor bug, and adds a couple of enhancements. Includes source in C. Author: Paul Thomas Miller

KeyMac A keyboard macro processor that allows you to record keystrokes and then play them back. V0.1,Source in Modula-II. Author: Fridtjof Siebert

MegaWB A program that makes it possible to make your WorkBench screen as large as you like. Version 1.2, Source in Modula-II. By: Fridtjof Siebert

MuchMore Another program like "more", "less", "pg", etc. This one uses its own screen to show the text using a slow scroll. Includes built-in help, commands to search for text, and commands to print the text. Works with PAL or NTSC, in normal or overscan modes. Supports 4 color text in bold, italic, underlined, or inverse fonts. Version 2.5, this is an update to version 1.8 from disk 234. Includes source in Modula-II and assembly code. Author: Fridtjof Siebert

MultiSelect A small program that makes it possible to select several icons on the workbench without having to press the Shift key. This is version 1.0, includes source in Modula-II. Author: Fridtjof Siebert

PowerPacker A shareware command and data cruncher, with a full intuition user interface. Executable files which are crunched are automatically decrunched upon execution, thus saving precious disk space on systems without hard disks. Version 2.3b, binary only. Author: Nico Francois

WBShadow A small program that creates a shadow for everything that's displayed on your WorkBench (Windows, Icons, Gadgets, Menus, Texts, etc). V 1.0, Source in Modula-II. Author: Fridtjof Siebert

With CLI command that allows you to start any other CLI command several times and give it all the files that match a file pattern as an argument, one at a time. V 1.0, Modula-II Source. By: Fridtjof Siebert

Fred Fish Disk 254

Etale Another "more" or "less" type file reader, which differs from the others in that it supports superscripts and subscripts at very low memory cost, and supports a complete Greek alphabet. Includes source. Author: William Hammond

MachII A "mouse accelerator" program that also includes hotkeys, the features of sun mouse, clicktofront, popcli, title bar clock with a bbs online charge accumulator, and more. This is version 2.6, an update to version 2.4c on disk 163. Binary only. Author: Brian Moats

PatternLib A sharable library which implements AmigaDOS pattern matching, and a program that compiles "fd" files into library headers and interface stubs. Includes source in Draco. Author: Chris Gray (?), posted on BIX by Jeff White

SlotCars A fast action one or two player game of "dueling" slotcars. Shoot rockets at your opponent and avoid his. Binary only. Author: Brian Moats

Uedit Version 2.5b of this nice shareware editor. Has learn mode, a command language, menu customization, hyper text, and other user configurability and customizability features. Binary only, shareware, replaces version 2.4g on disk 189. Author: Rick Stiles

For PDS orders, please use form on page CIII
Visa and MasterCard is available on orders of $20.00 or more.

Fred Fish Disk 255

CyclicSpace A Griffeathian cyclic space generator program. Implements a cellular automaton as described in the August 89 issue of Scientific American. V 1.42, includes source in assembly code. Author: A. J. Brouwer

MakeIcon This program will make an icon (.info file) using image data from either another icon, or an IFF picture file. The image data is drawn on the screen so you can see exactly what your icon will look like. You can also go the other way, and make IFF files from icons, which can then be loaded into a standard IFF graphics editor. Version 1.1, binary only, source available from author. Author: Paul Blum

MED A music editor much like SoundTracker. A song consists of up to 50 blocks of music, which can be played in any order. Editing features include cut/paste/copy tracks or blocks, changing the vibrato, tempo, crescendo, and note volume. Other features include switching of the low-pass-filter on or off on a per song basis, and a cute little animated pointer of a guy doing "jumping jacks" in time to the music! This is version 1.12, binary only. Source for a player program included. Author: Teijo Kinnunen

RolandD110 A program to transfer sound samples between the Amiga and a Roland D-110. Version 1.0, binary only. Author: Dieter Bruns

RolandS220 A program to transfer sound samples between the Amiga and a Roland S-220. This is version 2.0, an update to MIDIsoft version 1.0 on disk 199. Includes several enhancements. Binary only. Author: Dieter Bruns

SunMaze The 3d maze demo from disk 171, now expanded with shadows in the corridors to give more sense of direction. Includes source. Author: Werner Gunther, Shadows by Dirk Reisig

Fred Fish Disk 256

BlitDemons This program implements the Demons cellular automaton as described in the August, 1989, issue of Scientific American. Using extremely simple rules it exhibits rather complex behaviour. Uses the blitter to perform over 350,000 cell-generations per second. Version 1.0, includes source. Author: Walter Strickler

NameGame A game based on a popular type of word skill puzzle contest generally sponsored by newspapers. Binary only. Author: Mike Savin, Gladstone Productions

Stevie A public domain clone of the UNIX 'vi' editor. Supports window-sizing, arrow keys, and the help key. This is V3.7a, an update to FF217. Changes include the addition of ARP wildcard expansion for file names, support for the "!cmd" feature of vi, other small enhancements, and some bug fixes. Includes source. Author: Various, Amiga work by G. R. (Fred) Walter

Fred Fish Disk 257

ColorReq A color palette library callable from any program. Used to adjust the colors of any screen. Automatically adjusts to screen size, number of colors. Has many features including SPREAD, COPY, UNDO, RGB, HSV, as well as customizable gadgets. Binary only. Author: Jeff Glatt and Jim Fiore

FileIO An update to the file requester library that appeared on disk #203. This one has new features, include files, docs, and examples in C, assembly, and Basic. Binary only Author: Jeff Glatt and Jim Fiore based on original C by R.J. Mical

Vt VLT is both a VT100 emulator and a Tektronix (4014 plus subset of 4105) emulator, currently in use at SLAC (Stanford Linear Accelerator Center). Although the VT100 part was originally based on Dave Wecker et al.'s VT100, many enhancements were made. The program requires ARP, and it has an ARexx port. XMODEM 1K/CRC and Kermit protocol support also included. Other features include support for additional serial ports, external file transfer protocols, and "chat" mode. The Tektronix emulation allows saving IFF files, PostScript files, and printing bitmaps to the printer. This is version 4.226, an update to version 4.065 on disk 245 and version 4.036 on disk 226. It comes in two versions, one with Tektronix emulation, and one without. Other changes include full XPR version 2.0 support, and a scrollback/review/history buffer. Binary only. Author: Willy Langeveld

Fred Fish Disk 258

Backup Backup and Restore allow you to backup any directory tree with optional compression, and later extract all or part of the tree. The protection, date, and file comment are saved with each file. This is V2.04, an update to FF168. Binary only. Author: Matt Dillon

DMouse A versatile screen & mouse blanker, auto window activator, mouse accelerator, popcli, pop window to front, push window to back, etc, widget. Includes DLineArt, a screen blanker replacement program for use with DMouse. This is DMouse version 1.20, an update to the version on disk 238 (also called version 1.20). Includes source. Author: Matt Dillon

STReplay A shared library which allows loading and playing of SoundTracker modules, even from high level languages like Basic or C, without any special effort. Includes source. Author: Oliver Wagner

Suplib The support library needed to rebuild various programs of Matt's from the source, including Dme, DMouse, etc. Update to FF169. Source. By: Matt Dillon

Fred Fish Disk 259

EFJ A fast action machine-code game (Escape From Jovi) featuring hi-res scrolling, large playfield, disk based high score list, stereo sound, multiple levels, a cheat mode, realistic inertia and gravitational effects, an animated sprite, and more. Use a joystick in port 2 to control the ship. This is version III, an update to the version on disk 148. Includes source in assembly code. Author: Oliver Wagner

Fred Fish Disk 260

Accordion Demo version of a solitaire type card game. The object of the game is to condense all the cards into one pile. The cards are dealt face up from left to right, one at a time. Piles are built by moving a card or pile from the left onto a card or pile to the right. V 1.0.1, binary only. Author: Steve Francis

Calculation Demo version of a solitaire type card game. The object of the game is to stack the cards into four ordered stacks, one that counts by one, another that counts by two, another that counts by three and another that counts by four. Version 1.0.2, binary only. Author: Steve Francis

CCLib An implementation of the standard C library, that is done as an Amiga shared library, containing over 140 functions. Version 1.3, binary only. Author: Robert Albrecht

SeaHaven Demo version of a solitaire type card game. The object of the game is to separate the deck into its four suits, each in an ordered stack from ace to king. Version 1.1, binary only. Author: Steve Francis

Fred Fish Disk 261

CopDis A copper list disassembler that can be run from the CLI or linked with and run directly from an application program. This is version 0.0a and includes source. Author: Karl Lehenbauer

ShowBiz A fun animation with mice, a dancing alligator, a sheep on a trampoline, and more. Includes source in "director format". Author: Robert Corns

XprZmodem An Amiga shared library which provides ZModem file transfer capability to any XPR-compatible communications program. This is version 2.0, an update to version 1.0 on disk 236. It adds support for XPR spec version 2.0 capabilities, including automatic download activation, better user interface for setting options, improved transmission speed, and other minor enhancements. Includes source. Author: Rick Huebner

Fred Fish Disk 262

Indent This is version 1.1 of GNU Indent, a C source code formatter/indenter. Especially useful for cleaning up inconsistently indented code. Includes source. Authors: Developed by the University of California, Berkeley, the University of Illinois, Urbana, and Sun Microsystems, Inc. GNU version by Jim Kingdon, Amiga port by Dan Riley.

Jumble A quick "brute force" program for solving for anagrams. Not very elegant, but gets the job done. Includes source. Author: Mike Groshart

Lotto Lotto is designed to replace your user group's old shoebox full of membership numbers and names and add some pizzaz to the process of drawing for doorprizes at club meetings. Includes source. Author: Mike Groshart

QMouse A very small mouse accelerator (4K) written in assembly language, but with most of the features of its larger cousins. Not related to QMouse on disk 49. This is version 1.6, shareware, binary only. Author: Lyman Epp

QView A very small file view program (only 3K) written in assembly language, but with most of the features of its larger cousins. Version 1.1, shareware, binary only. Author: Lyman Epp

WorldDataBank A project using geographical data, declassified and made available by the CIA under the Freedom of Information Act, to draw a Mercator projection of any area of the Earth that you would like to inspect, in various degrees of magnification. Also includes a program that displays a "satellite view" of any region. Version 2.0, includes source. Author: The CIA, Mike Groshart, Bob Dufford

Fred Fish Disk 263

Sentinel A large ray-traced animation created with Turbo Silver SV, Deluxe Photolab, Deluxe Paint III, and the Director. It consists of approximately 60 to 70 frames of animation, rendered during a period of about three and a half weeks. This is Bradley Schenck's entry to the 1989 BADGE Killer Demo contest, and won first place in the contest. Because of its size, it has been split across two disks. The rest of the files are on disk 264. Requires 3Mb or more of memory to run. Author: Bradley Schenck

Fred Fish Disk 264

Sentinel A large ray-traced animation. This is Bradley Schenck's entry to the 1989 BADGE Killer Demo contest, and won first place in the contest. Because of its size, it has been split across two disks. The rest of the files are on disk 263. Author: Bradley Schenck

Fred Fish Disk 265

VAUX-Killer A very funny animation which won 2nd place in the 1989 BADGE Killer Demo Contest. Because of its size, the demo is split across two disks. The other files are on disk 266. Two floppy drives, or a hard disk, are strongly recommended. Binary only. Author: Roger Curren

Fred Fish Disk 266

VAUX-Killer A very funny animation which won 2nd place in the 1989 BADGE Killer Demo Contest. Because of its size, the demo is split across two disks. The other files are on disk 265. Two floppy drives, or a hard disk, are strongly recommended. Binary only. Author: Roger Curren

Fred Fish Disk 267

Diglib An Amiga device independent graphics library for fortran applications. This is an enhanced and debugged version of a public domain library, the development of which was sponsored by the US Government. This library is required for part of the Matlab package, also included on this disk. Includes source in FORTRAN. Author: Hal Brand, Craig Wuest, James Locker

Mackie A versatile cli/macro-key initiator based on POPCLI with a unique method of "screen-blanking". I won't say more, just try it! This is version 1.3, an update to version 1.2 from disk 189. Now includes automatic generation of different patterns and some bug fixes. Includes source. Author: Tomas Rokicki

Matlab A FORTRAN package (MATrix LABoratory) developed by Argonne National Laboratories for in house use. It provides comprehensive vector and tensor operations in a package which may be programmed either through a macro language or through execution of script files. Supported functions include sin, cos, tan, arcfunctions, upper triangular, lower triangular, determinants, matrix multiplication, identity, hilbert matrices, eigenvalues, eigenvectors, matrix roots, matrix products, inversion, and more. Amiga specific features include workbench startup, polar plots, contour plots, enhanced plot buffer control, and algorithmic plot display generation. Includes source in FORTRAN. Author: Cleve Moler, Jim Locker

Fred Fish Disk 268

Doctor_A This animation of Amiguy on the basketball court is Marvin's entry to the 1989 BADGE Killer Demo Contest. The animation source (for SculptAnimate-4D) is available from the author. Author: Marvin Landis

Klide This line art demo is Jerry's entry to the 1989 BADGE Killer Demo Contest, where it won 4th place. Klide uses line mode blitter code which is capable of rendering short vectors at a rate up to 15,000 lines/second. Binary only. Author: Jerry Kallaus

OnlyAmiga This demo is Rob's entry to the 1989 BADGE Killer Demo Contest, where it won 9th place. It demonstrates sound, multitasking, HAM color, the blitter, and more. Binary only. Author: Rob Peck

Fred Fish Disk 269

Calendar A program that generates calendars in any one of 10 standard formats for any year after 1900. Binary only. Author: Pierre A du Parte

ChessTutor Chess tutor is a program designed to introduce novice chess players to the basics of the game. Shareware, written in AmigaBASIC. Author: William Jordan

PropGadget Example code for using proportional gadgets, written in assembly code and C, that can be called from your own application. Includes source. Author: Jerry Trantow

RadBoogie This demo, which won 7th place in the 1989 BADGE Killer Demo Contest, uses almost all features of the Amiga extensively, including the copper, blitter, sprite hardware, 68000 machine language, and preemptive prioritized multitasking. Includes source. Author: Dave Quick, Mark Riley, Tomas Rokicki

ShowDisk A useful program that graphically shows the map of sectors used on floppy drives by one or more files. The mapping is color coded so you can identify what sectors are used by the various directories and files. Includes source in assembly language. Author: Bernhard Meisner

SpinPointer While going through some musty archives I found this little gem that got overlooked before. It is a short source module that provides a "busy" indicator replacement for the standard mouse pointer. Includes source. Author: Mark Rinfret

Fred Fish Disk 270

ColumnSet A text filter program that takes as input a file with one word per line and produces a file with these words laid out in the same order in even columns, as many as will fit across the output screen or page with at least one space between columns. Includes source in Modula-2. Author: Kent Paul Dolan

MRBackUp A hard disk backup utility that does a file by file copy to standard AmigaDOS floppy disks. Includes an intuition interface and file compression. This is version 3.3d, an update to version 2.4 on disk 170. Binary only. Author: Mark Rinfret

TooMuch3D This animation is Jim's entry to the 1989 BADGE Killer Demo Contest, where it won 10th place. It is a warning to those of you who spend too much time in front of your computer monitors. Binary only. Author: Jim Robinson

Fred Fish Disk 271

CPUStandoff This cute "computer wars" animation is Bob's entry to the 1989 BADGE Killer Demo Contest. Lots of visual jokes in this one. Binary only. Author: Bob Janousek

LaserBoing This ray-traced animation with sound is Mike's entry to the 1989 BADGE Killer Demo Contest. This "Amiga Boing" ball is just loaded with energy! Binary only. By: Mike van der Sommen

Fred Fish Disk 272

AmigaPunt A program designed to predict the performance of horses in a race. The premise is that the factors affecting a horse's performance, and ultimately the result of the race, can be given a value. Binary only. Author: Pierre A du Parte

Balloon This animation with sound effects is Dave's entry to the 1989 BADGE Killer Demo Contest. Binary only. Author: Dave Schreiber

Spigot This ray-traced animation, using overscan HAM and sound effects, is Eric's entry to the 1989 BADGE Killer Demo Contest– it won 6th place. Binary only. By: Eric J. Fleischer (Dr Gandalf)

Fred Fish Disk 273

BattleForce A nicely done shareware game, submitted by the author, that simulates combat between two or more giant, robot-like machines. This is version 3.61, an update to version 3.01 on disk 205. Binary only. Author: Ralph Reed

PennyWise An easy to use, flexible cashbook program using an Amiga Intuition interface. It can be used to keep track of the financial transactions of a cheque, bank, business, or similar account. Shareware, binary only. Author: Pierre A du Parte

Fred Fish Disk 274

HP11 Emulates an HP11C calculator including the program mode. Features an ON/OFF button that turns the calculator into an icon that will sit and wait until you need it again. This is version 1.1, an update to FF153, and includes a few bug fixes and minor changes. Binary only. By: David Gay

KeyBiz This little hack will make you think you have mice scurrying around in the back of your computer. Includes source. Author: Mark Schretlen

LookFor A general "find that file" utility which is handy for hard drives, multiple drives, and multilayered paths. Includes source. Author: Mark Schretlen

SlideMaster A slideshow program that can show any IFF ILBM picture, including HAM, extra half bright, hires, interlace, and overscan, using several different wipes. Also has an ARexx port and is fairly small. Version 0.1, binary only. By Aric Caley

Snap A tool for clipping text or graphics from the screen, using the clipboard device. Snap finds out character coordinates automatically, handles different fonts, keymaps, accented characters, and more. Version 1.3, includes source. Author: Mikael Karlsson

Xoper Very comprehensive program to monitor and control system activity. Monitor cpu, memory usage, ports, interrupts, devices. Close windows, screens, show loaded fonts or last Guru code number. Clean up memory, flush unused libraries, devices, fonts. etc. and a whole bunch more! Spawns its own process. A very handy background task to have loaded. This is version 2.0, an update to FF228, and has a completely rewritten interface and an iconify feature. Assembly source included. By Werner Gunther

Fred Fish Disk 275

Vt100 Version 2.9 of the original Amiga vt100 emulator with kermit and xmodem file transfer. This version adds an AREXX port, new script and AREXX commands, some bug fixes, the ability to use custom external protocol modules (not XPR), and support for zmodem. Update to version 2.8 on disk 138. Includes source. Author: Dave Wecker, Tony Sumrall, Frank Anthes, and Chuck Forsberg

Fred Fish Disk 276

Blit This animation is Richard's entry to the 1989 BADGE Killer Demo Contest. An interesting feature of this animation is that it uses the Copper to show the lower right corner of the screen in Hi-Res, while leaving the rest of the screen in Lo-Res. Binary only. Author: Richard Addison

ClickDOS A "directory utility" type program which is reasonably small, uses a single window on the Workbench screen (which can also be iconified), does not rely on other programs for most of its functions, and is very memory efficient. This is version 1.10, binary only. Author: Gary Scott Yates

DateRequester A module that provides Intuition based support for soliciting a date value from the user. It supports both point-and-click selection of date values and direct entry of the individual components. It also includes a standalone ARexx driven date requester program which can be called from ARexx macro files. Includes source. Author: Mark Rinfret

Fred Fish Disk 277

ARTMARTM (Amiga Real Time Monitor) displays and controls system activity such as tasks, windows, libraries, devices, resources, ports, residents, interrupts, and vectors. Version 0.9, binary only. Author: Dietmar Jansen and F. J. Mertens

Icon A high-level programming language with extensive facilities for processing strings and lists. Icon has several novel features, including expressions that may produce sequences of results, goal-directed evaluation that automatically searches for a successful result, and string scanning that allows operations on strings to be formulated at a high conceptual level. Icon resembles SNOBOL4 in its emphasis on high-level string processing and a design philosophy that allows ease of programming and short, concise programs. This is version 7.5 of the public domain implementation of Icon from the University of Arizona. It is an update to version 6.0 from disk 81. Binary only. Author: Ralph Griswold, Clinton Jeffery, et. al.

LabelPrint A program that allows you to easily print labels for your disks. This is version 3.0, an update to version 2.5 from disk 238. Shareware, binary only (source available from author). By: Andreas Krebs

LPE LaTeX Picture Editor is a graphical editor for producing "pictures" for the LaTeX system, which may be imported by LaTeX. You can draw boxes, dashed boxes, lines, vectors, circles, boxes with centered text, and plain text. This is version 1.1, an update to version 1.0 on disk 243, and includes some new drawing tools, enhanced user interface features, optimized code, and some bug fixes. Binary only. Author: Joerg Geissler

BIT MOVIE '90

COMPETITION FOR COMPUTER ANIMATION IN REAL TIME

The Adriatic Coast Amiga User Club and the Municipality of Riccione Italy have organized the Third Competition for Computer Animation in Real Time.

1. All works rendered in real time on the Amiga will be accepted.
2. Works will be accepted which use any and all programs available, however, no works will be accepted which use public domain objects or backgrounds.
3. Each competitor may submit up to four works.
4. The animation must be submitted on floppy disk and there must be a description and script for each.
5. For each animation, there must also be a still picture on disk that can be used for a photograph.
6. All submissions must be sent to: Adriatic Coast Amiga User Club—c/o Carlo Mainardi—Via Bologna n.13—47036 Riccione (FO) Italy.
7. A special commission will decide which works will be admitted to the competition.
8. Deadline for all materials is March 15, 1990.
9. The winning animations will be selected by the show's attendees and a qualified panel All decisions are final.

Prizes:

First Prize	Lit.1,000,000
Second Prize	Lit. 600,000
Third Prize	Lit. 400,000

Contest information and rules were translated (to the best of our ability) from a letter which we received from Carlo Mainardi. According to the letter, over 1500 people came to last year's competition. All entries last year were from Italy. Judging will take place April 14-16, 1990. For more information, write to:

Adriatic Coast Amiga User Club
c/o Carlo Mainardi
Via Bologna n.13
47036 Riccione (FO) Italy

WBAssign A small WorkBench program that creates AmigaDOS assignments without having to open a CLI. Assignments may be made by specifying a complete path (as supported by the CLI command), or by specifying a path relative to the current directory. This allows a program and its assigns to be moved easily. In addition, WBAssign can create assigns specified in the icon files of other programs located anywhere in the system. This is version 1.20, binary only. Author: John Gerlach Jr.

WBRes A program that allows WorkBench users to have resident programs, as in the resident capability of WShell, the ARP shell, and the WB1.3 Shell. Version 1.2b, shareware, binary only. Author: John Bickers

Fred Fish Disk 278

AmigaTrekThree stories which are a parody of the Star Trek series, with an Amiga flavor. Author: Mike Smithwick

ASpice A version of the SPICE 2G.6 circuit analysis program which has been modified to run in the Amiga environment. The program arrays are adjusted to require one tenth the memory of the DEC VAX version. Although this does not usually put much of a constraint on circuit analysis, some users who are used to the full mainframe environment may have to be more aware of the memory demands of their analysis. Requires a minimum of 1 MB memory. This version neither supports nor requires the 68020 processor or 68881 coprocessor. This is Amiga version 5.1, an update to FF177. Binary only. Authors: Many, see Documentation. Amiga port by Dan Warns.

Frag Two CLI utilities that show disk and file fragmentation on AmigaDOS floppies. Includes source. By: David Gay

Fred Fish Disk 279

Ash A ksh-like shell for the Amiga. Some of its features include command substitution, shell functions with parameters, aliases, local variables, local functions, local aliases, powerful control structures and tests, emacs style line editing and history functions, I/O redirection, pipes, large variety of built-in commands, Unix style wildcards, Unix style filename conventions, filename completion, and coexistence with scripts from other shells. Very well documented. Version 1.0, binary only. Author: Steve Koren

MouseClock A clock and free memory display utility that generates a small display tied to your mouse, using two hardware sprites as the display area. Version 1.2, includes source. Author: Olaf Barthel

MRBackUp A hard disk backup utility that does a file by file copy to standard AmigaDOS floppy disks. Includes an intuition interface and file compression. This is version 3.3e, an update to FF270, to fix a serious bug in that version. Binary only. By: Mark Rinfret

MRPrint A detabbing print utility that sends text files to either the printer device or the standard output. Besides expanding tabs, it will also generate page headers, line numbers, and new margins. Version 3.4, includes source. By: Mark Rinfret

Fred Fish Disk 280

Berserker A virus detection program that can detect various forms of the common bootblock and link type viruses. Rather than checking for a specific virus, it looks for conditions that indicate a possible virus infection, thus it can detect new strains of similar viruses. Version 3.0+, includes source in assembly. Author: Ralf Thanner

CM A celestial mechanics simulator with an Intuition interface. You construct a layout of celestial bodies and specify various parameters for the bodies and the simulation. CM then animates the bodies according to the laws of gravitational attraction. Setups may be saved to disk to later reinact interesting scenarios. Version 1.0, includes source. Author: W. John Guineau

Graph A program that draws mathematical functions on a plane. Text and axes may be added, and the result saved to disk or printed. Uses a full Intuition interface with multiple windows (one per graph), requesters, etc. Version 1.0, includes source. Author: David Gay

Fred Fish Disk 281

Diff GNU diff and diff3 utilities, version 1.10. This version provides all the features of BSD's diff plus options to diff non-ASCII files, to ignore changes that just insert or delete blank lines, to specify the amount of context for context diffs, plus more. This version of GNU diff is part of the RCS package found on FF282. Includes source. By: Mike Haertel, David Hayes, Richard Stallman & Len Tower. Amiga port by Raymond Brand & Rick Schaeffer.

DiffDir Compares the contents of two directories, generating a list of differences to the standard output. Detects files or directories present in one hierarchy but not the other, file attributes that are different (dates, flags, comments, etc), and differences in file size. Version 1.1, an update to FF188. Includes source. Author: Mark Rinfret

FarPrint Debugging functions for programs which don't have any links to their environment. FarPrint consists of two major parts; a harbour process open to receive and distribute messages and requests, and a set of C functions to be linked into any program wishing to communicate with the FarPrint main process. Version 1.3, includes source. Author: Olaf Barthel

LoadImage An IFF ILBM reader that accepts overscanned pictures, allows you to scroll around in the bitmap if the picture is larger than the current display, works on both PAL and NTSC machines, supports color cycling using interrupt code, and supports printing of image portions. Version 1.9, includes source. Author: Olaf Barthel

MRARPFile Extended ARP file support package that adds an orthogonal set of routines which support generalized I/O with resource tracking while using the ARP library. Includes source. Author: Mark Rinfret

MRMan A package which contains an Amiga document reading system similar to the "man" command on UNIX systems. It offers a high degree of flexibility in the naming and placement of document files on your system. This is version 1.0 and includes source. Author: Mark Rinfret

Fred Fish Disk 282

PrintHandler A custom PRT: driver which offers easy single sheet support as well as limited data spooling. Version 1.1, includes source. Author: Olaf Barthel

RCS The Revision Control System (RCS) manages multiple revisions of text files. RCS automates the storing, retrieval, logging, identification, and merging of revisions. RCS is useful for text that is revised frequently, for example programs, documentation, graphics, papers, form letters, etc. This is RCS version 1.2, and includes source (the source to the GNU diff program used with this distribution of RCS can be found on disk 281). Author: Walter Tichy. Amiga port by Raymond Brand.

To Be Continued.....

In Conclusion

To the best of our knowledge, the materials in this library are freely distributable. This means they were either publicly posted and placed in the public domain by their authors, or they have restrictions published in their files to which we have adhered. If you become aware of any violation of the authors' wishes, please contact us by mail.

IMPORTANT NOTICE!

This list is compiled and published as a service to the Commodore Amiga community for informational purposes only. Its use is restricted to non-commercial groups only! Any duplication for commercial purposes is strictly forbidden. As a part of Amazing Computing™, this list is inherently copyrighted. Any infringement on this proprietary copyright without expressed written permission of the publishers will incur the full force of legal actions.

Any non-commercial Amiga user group wishing to duplicate this list should contact:

PiM Publications, Inc.
P.O.Box 869
Fall River, MA 02722

PiM Publications Inc. is extremely interested in helping any Amiga user groups in non-commercial support for the Amiga.



For PDS orders, please use form on page CIII
Visa and MasterCard is available on orders of $20.00 or more.

Amazing COMPUTING™ Your Original AMIGA® Monthly Resource for the Commodore AMIGA & AC's GUIDE to the Commodore AMIGA

Two Great New Ways to SAVE on the Original Amiga Monthly Resource

Complete Today, or Telephone 1-800-345-3360

Name_____

Address_____

City_____State_____Zip_____

Charge my ☐Visa ☐MC #_____

Expiration Date_____Signature_____

MasterCard® VISA®

All Charges are subject to a $20.00 minimum (charges under $20.00 will receive a $2.00 service charge).

Please circle if this is a **New Subscription** or a **Renewal**

PROPER ADDRESS REQUIRED. In order to expedite and guarantee your order, all large Public Domain Software orders, as well as most Back issue orders, are shipped by United Parcel Service. UPS requires that all packages be addressed to a street address for correct delivery.

One Year Of Amazing!	Our regular renewal now includes 12 monthly issues of **Amazing Computing plus** the Spring edition of AC's GUIDE to AMIGA	☐ $28.00 U.S ☐ $44.00 Foreign Surface ☐ $36.00 Canada and Mexico
Our New SuperSub!	**12** monthly issues of **Amazing Computing PLUS** AC's GUIDE to AMIGA **3** Product Guides! Spring, Fall, and Winter! A savings of $32.25 off the newsstand price. The Product Guides alone retail for $6.95 each	☐ $36.00 U.S ☐ $52.00 Foreign Surface ☐ $44.00 Canada and Mexico

Please circle any additional choices below: *(Domestic and Foreign air mail rates available on request)*

Back Issues: $5.00 each US, $6.00 each Canada and Mexico, $7.00 each Foreign Surface.

V1.1 V1.2 V1.3 V1.4 V1.5 V1.6 V1.7 V1.8 V1.9 V2.1 V2.2 V2.3 V2.4 V2.5 V2.6 V2.7 V2.8
V2.9 V2.10 V2.11 V2.12 V3.1 V3.2 V3.3 V3.4 V3.5 V3.6 V3.7 V3.8 V3.9 V3.10 V3.11 V3.12 V4.1
V4.2 V4.3 V4.4 V4.5 V4.6 V4.7 V4.8 V4.9 V4.10 V4.11 V4.12

Public Domain Software:

$6.00 each for subscribers (yes, even the new ones!)

$7.00 each for non subscribers (three disk minimum on all foreign orders)

Please circle your Public Domain Software choices below:

Amazing on Disk: A#1. . .Source & Listings V3.8& V3.9 A#2. . .Source & Listings V4.4
A#3. . .Source & Listings V4.5 & V4.6 A#4. . .Source & Listings V4.7 & V4.8
A#5. . .Source & Listings V4.9 A#6. . .Source & Listings V4.10 & V4.11

InNOCKulation Disk: IN#1...Virus protection A#7. . .Source & Listings V4.12 & V5.1

AMICUS

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26

Fred Fish Disks

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38
39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 NA
58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76
77 78 79 NA 81 82 83 84 85 86 87 NA 89 90 91 92 93 94 95
96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114
115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133
134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152
153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171
172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190
191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209
210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228
229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247
248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266
267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282

(NA Denotes disks removed from the collection)

Subscription: $_____

Back Issues: $_____

PDS Disks: $_____

Total: $_____

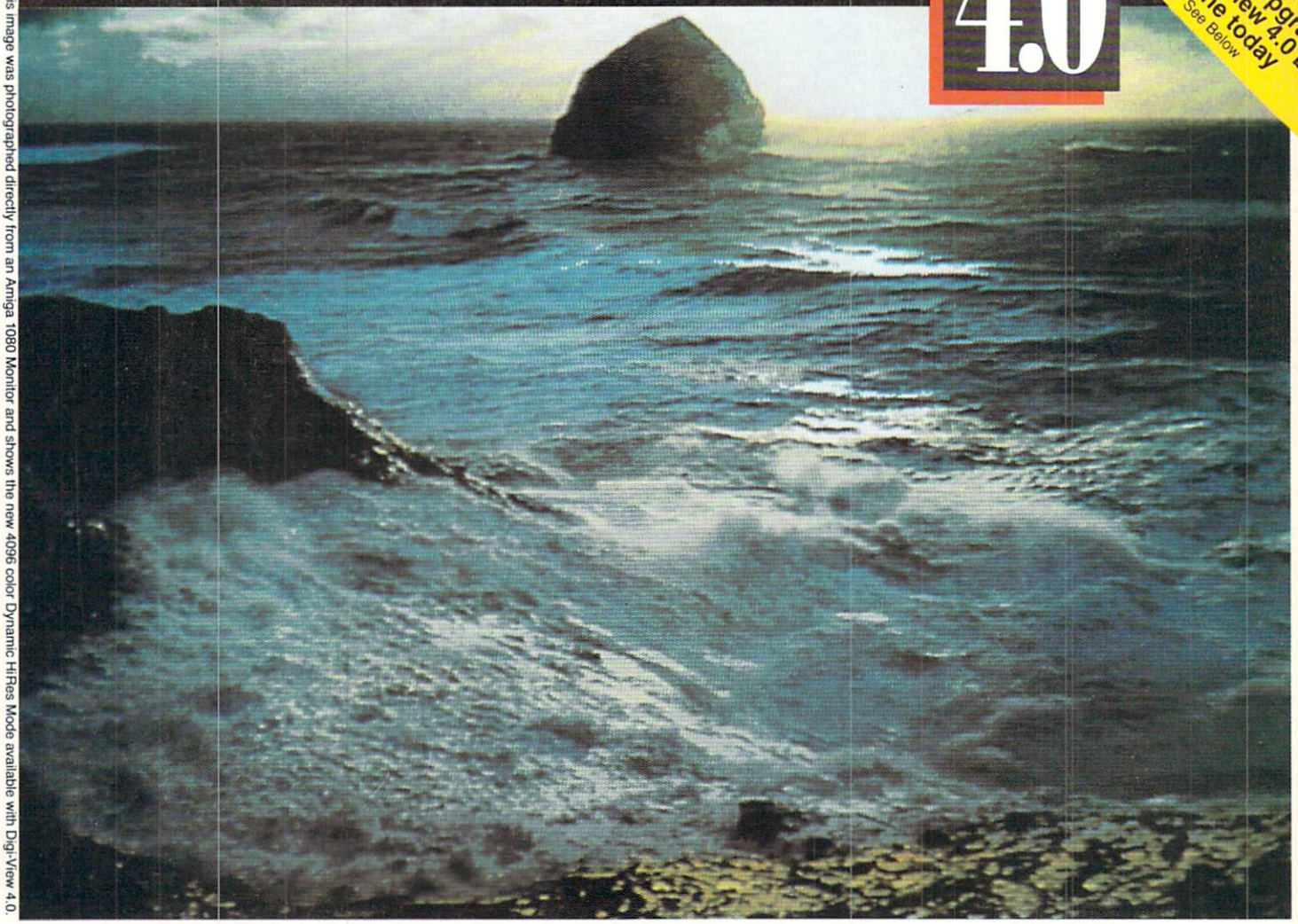
Please complete this form and mail with check, money order or credit card information to:

PiM Publications, Inc.
P.O. Box 869
Fall River, MA 02722-0869

Please allow 4 to 6 weeks for delivery

DIGI·VIEW GOLD 4.0

Attention All Digi-View Owners: Upgrade to Digi-View 4.0 by phone today
See Below



This image was photographed directly from an Amiga 1080 Monitor and shows the new 4096 color Dynamic HiRes Mode available with Digi-View 4.0.

4096 Colors in High Resolution

Digi-View Gold, the best video digitizer for the Amiga, just got one hell of a lot better. With the all new Digi-View 4.0 software you can do the one thing most people thought was impossible—digitize and display all 4096 colors in high resolution! We call this revolutionary new graphics mode Dynamic HiRes and you will have to see it on your own screen to believe it. But that's just the start of what makes Digi-View 4.0 a breakthrough. Other new features include: Dynamic HAM (fringe free HAM), Noise Reduction (for the sharpest images ever), ARexx support, super bitmap digitizing (directly into Digi-Paint 3), 24 bit color support, 68020 compatibility and dozens of other new features making Digi-View Gold the hottest Amiga graphics product ever.

It's still just as easy to use as before. Simply focus your video camera on any object or picture, and in seconds Digi-View Gold turns it into Amiga graphics that glow with vibrant color and stunning clarity. Whether you are creating graphics for desktop publishing, presentations, video, or just for fun, Digi-View Gold gives you dazzling images with amazing simplicity.

Digi-View Gold is designed specifically for the Amiga 500, 2000, and 2500, and plugs directly into the parallel port. With complete software control of color saturation, brightness, sharpness, hue, resolution, and palette, advanced image processing is as easy as adjusting the controls on your television.

If you want the best graphics possible for your Amiga, then you need the best selling video digitizer of all time: Digi-View Gold.

Only Digi-View Gold:

- Can digitize in all Amiga resolution modes from 320x200 up to 768x480 (full-HiRes overscan)
- Uses 2 to 4096 colors (including extra halfbrite)
- Uses exclusive Dynamic modes for 4096 colors in HiRes
- Is 100% IFF compatible and works with any graphics software
- Can digitize in 21 bits per pixel (2.1 million colors) for the highest quality images possible
- Has advanced dithering routines that give an apparent 100,000 colors on screen simultaneously
- Has powerful image processing controls for complete IFF picture manipulation

Digi-View and Digi-View Gold owners easy upgrade: Upgrade to the incredible new Digi-View 4.0 software just by sending your name, address and $24.95 + $6.00 for shipping and handling ($30.95 total) to NewTek Digi-View 4.0 upgrade, 115 West Crane St., Topeka, KS 66603. If you have a Visa or Mastercard call our toll free order line at 1-800-843-8934 and we'll rush your upgrade to you immediately.

Only $199.95

Digi-View Gold is available now at your local Amiga dealer.
Or call 1-800-843-8934.

NewTek INCORPORATED

*Requires standard gender changer for use with Amiga 1000. Digi-View Gold, Digi-Paint 3 and Video Toaster are trademarks of NewTek, Inc. Give a man a VCR and he can watch reruns, give him a Video Toaster and he can make his own!

Circle 140 on Reader Service card.